CONTENTS

NASA Contractor Report 2862

Linear Systems Analysis Program, L224(QR)

Volume II: Supplemental System Design and Maintenance Document

K. W. Heidergott

CONTRACT NAS1-13918
NOVEMBER 1979

NASA

NASA Contractor Report 2862

Linear Systems Analysis Program, L224(QR)

Volume II: Supplemental System Design and Maintenance Document

K. W. Heidergott
Boeing Commercial Airplane Company
Seattle, Washington

Prepared for
Langley Research Center
under Contract NAS1-13918

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1979

BLANK PAGE

CONTENTS

TABLES

FIGURES

1.0 SUMMARY

This document describes the structure, design, and programming details of the QR program. Use of the program is described in volume I of this document. QR can be used as a standalone program or as a module of a program system called DYLOFLEX (see fig. 1) which was developed for NASA under contract NAS1-13918 (ref. 1).

1.1 PURPOSE AND OBJECTIVES

The QR program (L224) is primarily a tool for applying classical control systems analysis and synthesis techniques. To make use of the program, the physical system to be analyzed is initially described by a system of simultaneous ordinary linear differential equations with constant coefficients. By manipulating the input matrix of equations, sets of eigenvalues that represent the denominator (POLES) and numerator (ZEROS) of the system can be calculated. From the transfer functions, studies of time response, frequency response, and root locus may be performed. In addition to the aforementioned analysis techniques, two unsteady aerodynamics analyses options (unsteady frequency response and V-g analysis) are available.

1.2 PROGRAM DESIGN AND STRUCTURE

The QR program can be divided into five different analyses: eigenvalue rooting, time response, frequency PSD response, root locus, and unsteady aerodynamics. Each major computational task is generally built into one overlay of the program. Figure 2 illustrates card input, external magnetic file input, punched matrix card output and plot vector file output.

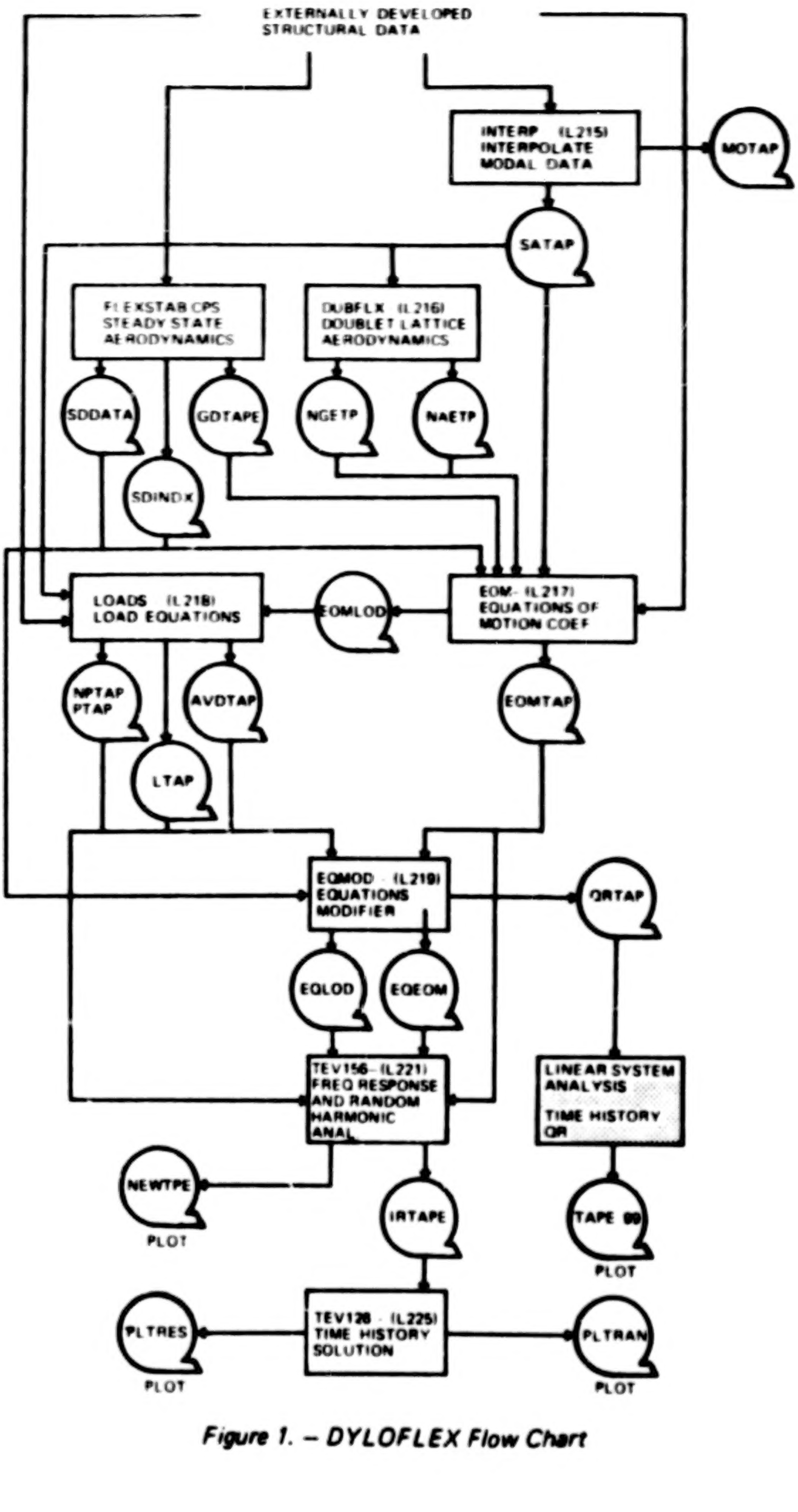


Figure 1. – DYLOFLEX Flow Chart

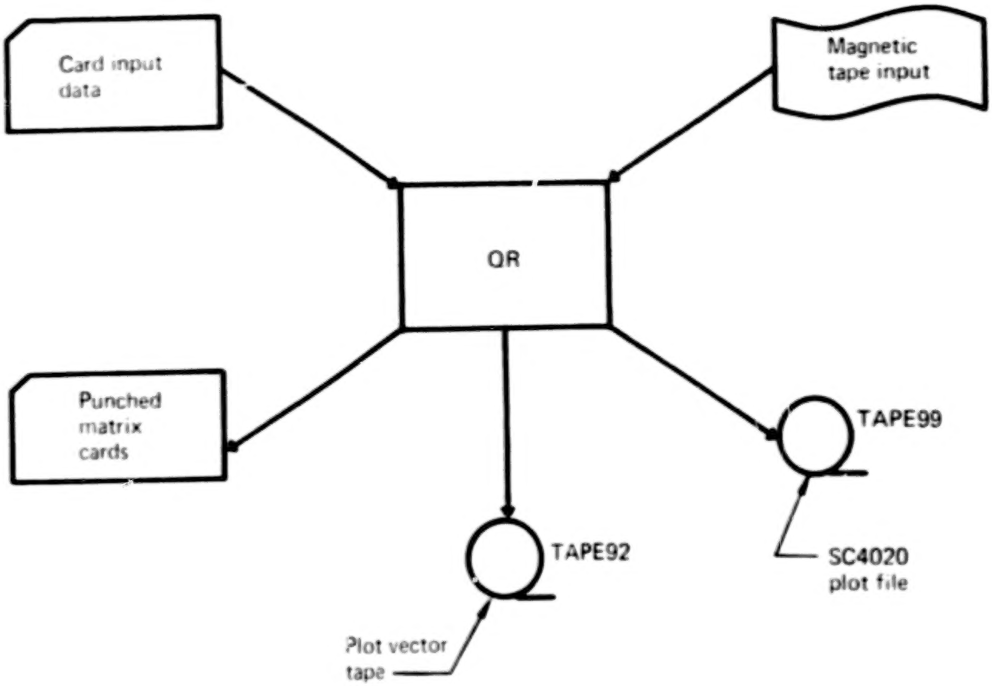


Figure 2. – QR Communication With External Files

2.0 PROGRAM DESCRIPTION

2.1 DISCUSSION

A fairly complete program description of QR can be found in sections 4.0, 5.0 and 6.0 in volume I of this document. Figure 3 shows a block diagram of the overlay structure in QR. The last letter of the overlay block name generally signifies what is actually done during execution. The M in QRM stands for main executive program (MAIN); T in QRT stands for time response analysis; R in QRR stands for root locus; U in QRU stands for unsteady aerodynamics. Sections 2.1.1 through 2.1.17 detail the task of each overlay.

2.1.1 OVERLAY QRM.0.0

This overlay is the QR main program and its purpose is to control the tasks that are undertaken. The control process is very straight forward. QR reads a data card from the INPUT file and determines if a valid command has been input. Assuming a valid command has been input, this overlay will do one of two things: (1) branch to another subroutine or position in the main program (QR) where additional input data is read or analytic steps are performed, (2) call another overlay to perform time response, frequency response, root locus plotting, V-g analysis plotting, or unsteady aerodynamics options.

The principal analytic step undertaken in the QRM.0.0 overlay is eigenvalue rooting. The technique used to reduce the polynomial input matrix and calculate eigenvalues is discussed in reference 2. In addition, this overlay: (1) controls looping for phase gain locus and calculation of open loop zero and poles (section 4.2, volume I discusses this calculation); (2) controls looping for altitude and reduced frequencies when performing V-g analysis (section 4.9 of volume I discusses the theory for V-g analysis); and (3) calculates the polynomial coefficients associated with a given set of eigenvalues; (section 4.4 of volume I discusses this calculation).

I/O devices are directly or indirectly opened in this overlay. They include:

(1) INPUT file or TAPE5

(2) OUTPUT file or TAPE6

(3) PUNCH file or TAPE93

(4) Interactive data base file or TAPE7

(5) Matrix storage file or TAPE90

(6) "In-core" matrix file or TAPE91

(7) Root locus and flutter eigenvalue storage file or TAPE96

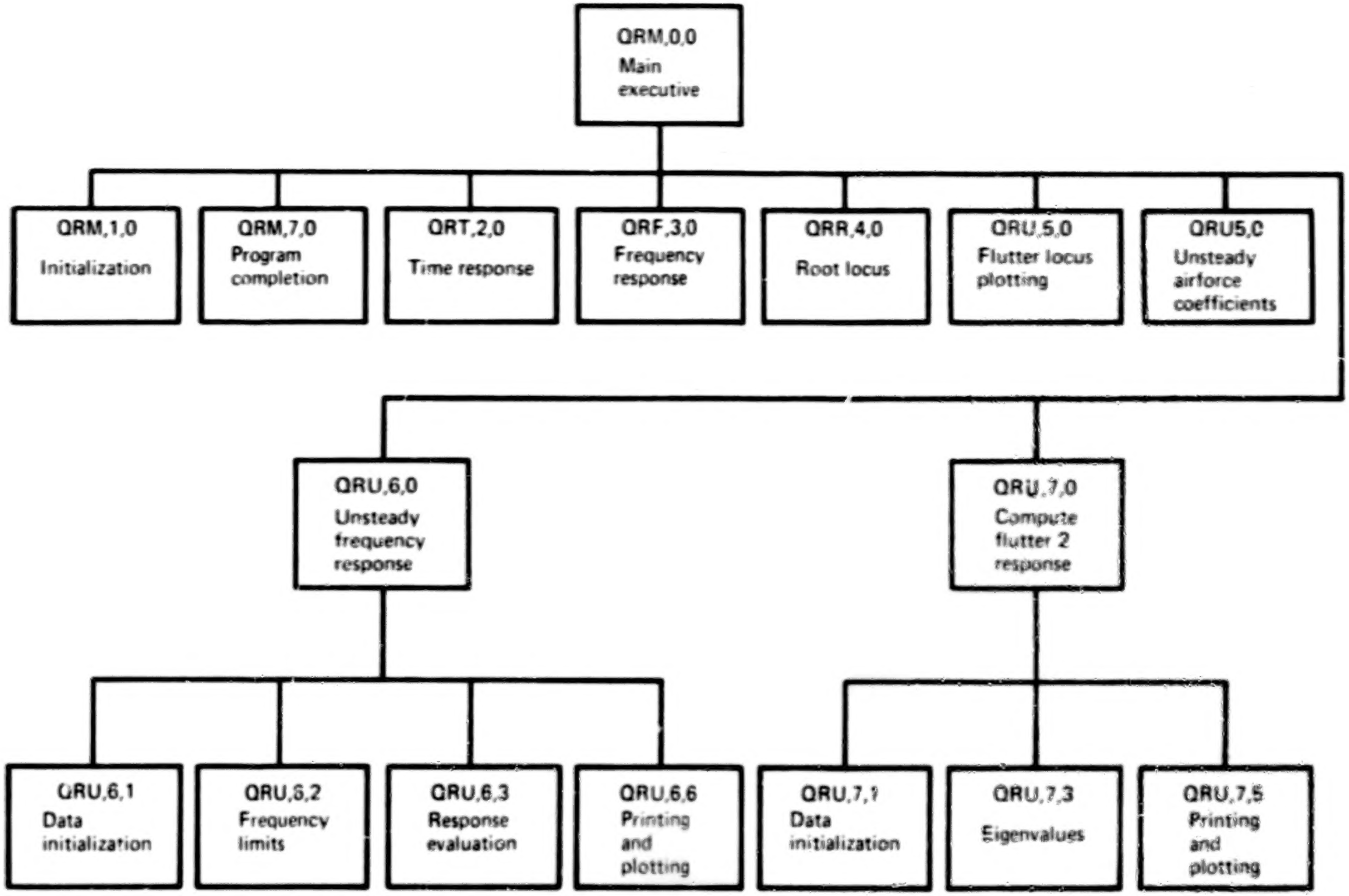


Figure 3. – QR Overlay Structure

(8) SC4020 plot output file or TAPE99

(9) Plot vector file or TAPE92

(10) READ MATRIX TAPE data files – TAPE1, TAPE2, TAPE3, TAPE4, TAPE8, TAPE10, TAPE11, TAPE12, TAPE13, TAPE14, TAPE15, TAPE16, TAPE17, TAPE18, TAPE19, TAPE20

Figure 4 is a macro flow chart for the QRM,0,0 overlay.

A map of subroutines called in QRM,0,0 follows (table 1).

2.1.2 OVERLAY QRM,1,0

This overlay calculates constants that are to be used in QR, prints an identification page on the output file (TAPE6), sets up pooled program buffers, and reads the first QR data card from the input file (TAPE5).

Table 2. – Subroutines Called by Overlay (QRM,1,0)

QRM10 Calls	BIGGER	
	COST*	
	KOMSTR†	
	EOF*	
	MUSER	
	TTY*	
	CLOCK*	
	WA10	
	RWA10	
	VUSER	
	FETAD†	
	LOCF*	
	STRMOV†	
	PRINTV – calls	A10R1
		BUFFIL
	USERSB	
	RA10	

† DYLOFLEX library routine

* CDC system library routine

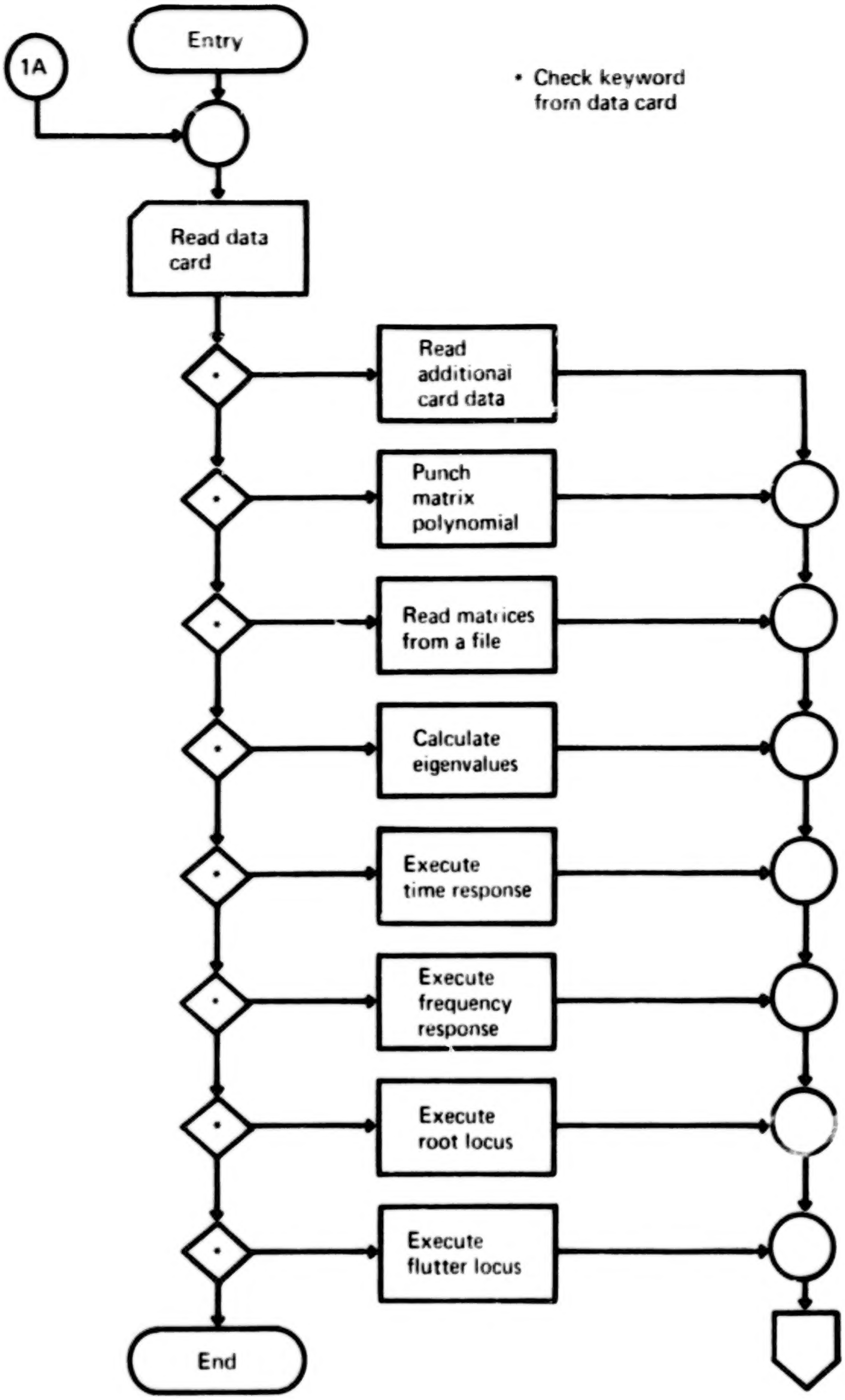


Figure 4. – Macro Flow Chart of Overlay (QRM, 0,0)

Table 1 — Subroutines Called by Overlay (QRM,0,0)

(QRM,0,0) Program QR				
ACCT	AUTOFL	REQFL†		
AUTOFL	REQFL*			
BEGINS	AUTOFL	REQFL†		
	BLKFLL			
	CLEAR			
	RWA10			
	WARN			
BLKFLL				
CANCEL	FORMPM			
	RPRINT	SECOND*		
		STRMOV†		
CLEAR				
COL				
COLROW	RW12	RUW (see below)		
	RWI5	RUW (see below)		
CRAMER	RWI5	RUW	COL	
			ENDQR	AUTOFL
			ISCAN†	
			KNVRT†	
			KOMPAR†	
			LOCF*	
			SCAMB	
			STRMOV†	

† DYLOFLEX library routine
* CDC FORTRAN library routine

Table 1. — (continued)

CREAD	CLEAR		
	CSAVE		
	ENDQR	AUTOFL	REQFL†
	QQQQ7		
	REGRUP		
	RUW (see above)		
	RWE12	RUW (see above)	
	RWI5	RUW (see above)	
CSAVE			
ENDQR	AUTOFL	REQFL†	
FORMPM			
FRESP	AUTOFL	REQFL†	
FRGRID	RWE12 (see above)		
	RWI5 (see above)		
FTPLOT	AUTOFL	REQFL†	
GAINDC			
GAINS	RUW (see above)		
	RWE12 (see above)		
	RWI5 (see above)		
GUST	RWE12 (see above)		
	RWI5 (see above)		
MUSER			
ORDER			
PGSET	RSTORE	AUTOFL	REQFL†
PNCH			
PZSTRE			

† DYLOFLEX library routine
* CDC FORTRAN library routine

Table 1. – (concluded)

QCKROO	AUTOFL	REQFL†		
	CLEAR			
	MULLP	CCOMPE*		
		CLDIV*		
		CREV*		
	SECOND*			
QRROOT (see figure 19)				
READEM	RDMAT	READTP†		
		TAPDIM		
		WRITER		
READPT	AUTOFL	REQFL†		
RLGRID	RWE12	RUW (see above)		
	RWI5	RUW (see above)		
RSTORE	AUTOFL	REQFL†		
RUW (see above)				
RWA10				
RWE12	RUW (see above)			
RWI5	RUW (see above)			
SENSOR	READTP*			
	RWE12	RUW (see above)		
	RWI5	RUW (see above)		
	TAPDIM	ENDOR	AUTOFL	REQFL†
		WARN		
TRPLOT	AUTOFL	REQFL†		
USERB				
VREAD	CLEAR			
	RUW (see above)			
VUSER				
WA10				
WRITUM	WRITER			
ZFIND	RSTORE	AUTOFL	REQFL†	

† DYLOFLEX library routine
* CDC FORTRAN library routine

2.1.3 OVERLAY QRM.7,0

This overlay is the last logical step executed in the QR program. The maximum field length used during the QR execution is calculated and printed on the output file (TAPE6) and the dayfile, the SC4020 plot buffer is filled and emptied onto disk unit 99, local scratch files TAPE90, TAPE91 and TAPE96 are returned, and the CRU cost of the QR execution is calculated.

Table 3. – Subroutines Called by Overlay (QRM,7,0)

	BUFOUT		COST*
	EXIT*		CLOCK*
	REMARK*		TTY*
QRM70 calls	QRCOST	calls	PFAPPND*
	WA10		GTACCID*
	RETURNF†		FETADD†
	BUFFIL		RETURNF†
			USERNUM*

† DYLOFLEX library routine
* CDC system library routine

2.1.4 OVERLAY QRT.2,0

This overlay calculates time response given a polynomial transfer function (ZEROS(s) POLES(s)), time intervals, and print plot instructions. All of this input information is read from data cards or calculated in the QRM0,0 overlay and passed into QRT.2,0 by named common.

Sections 4.3 and 4.8 of volume I discuss the analytical approaches used to solve time response options available in QR.

Once the time response has been calculated and stored in blank common, it is printed plotted using the specifications established by the user. Section 6.3.5 of volume I details time response data cards. The time delay for the COMPUTE TIME RESPONSE WITH TAU is read and written from input and output respectively in this overlay.

Figure 5 is a macro flow chart for the QRT.2,0 overlay.

Table 4 displays a map of subroutines called in QRT.2,0.

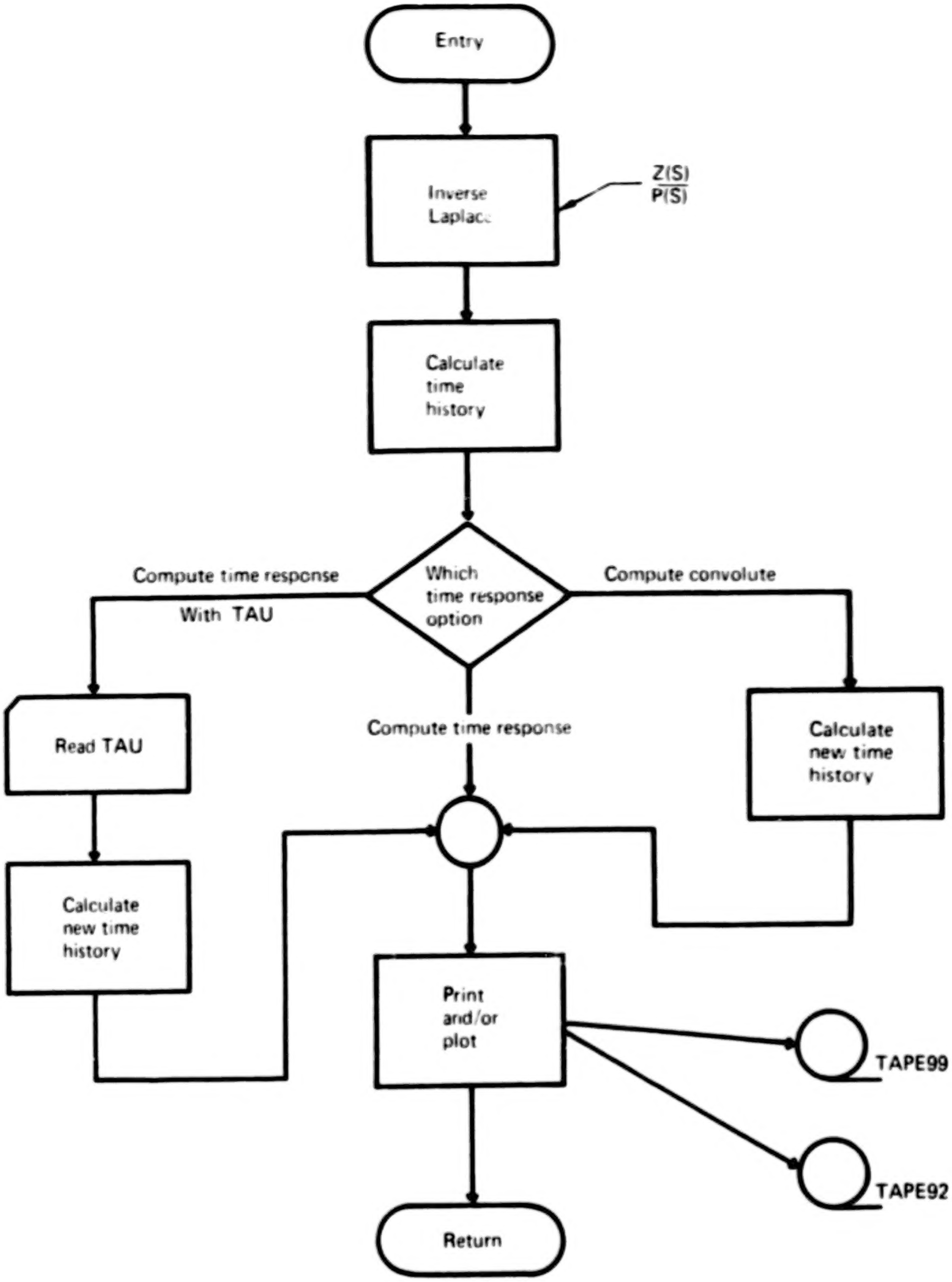


Figure 5. – Macro Flow Chart of Overlay (QRT, 2,0)

Table 4. – Subroutines Called by Overlay (QRT,2,0)

OVERLAY (QRT,2,0)
PROGRAM QRT20

- INLAP
  - CGLESM†
  - GLESOM†
- NEGMAT
- PLOTTM
  - CLEAR
  - KGRID
    - DXDYV*
    - GRID1V*
    - SETMIV*
  - PLTLI
    - IXV
    - IYV
    - LINEV
  - SCALE1
  - VPRNTD
    - A10R1
    - BUFFIL
  - WARN
  - WRTETP†
- PTPLOT
  - PRPLOT
    - BLKFLL
    - PRLING
  - SCALE2
- RWE12
- TBLU1†
- TPRINT

† DYLOFLEX library routine
* CDC system library routine

2.1.5 OVERLAY QRF.3.0

This overlay calculates frequency and PSD response given a transfer function (ZEROS(s) POLES(s)), gust spectrum, and print plot instructions. All of this input information is read from data cards or calculated in the QRM.0.0 overlay and passed into QRT.2.0 by named common.

Sections 4.6 and 4.7 of volume I discuss the analytical approaches used to solve for frequency and PSD response in the QR program.

Once the frequency or PSD response has been calculated and stored in blank common, it is printed plotted using the specifications established by the user. Section 6.3.6 of volume I details frequency response data cards.

Figure 6 is a macro flow chart for the QRF.3.0 overlay. Table 5 displays a map of subroutines called in QRF.3.0.

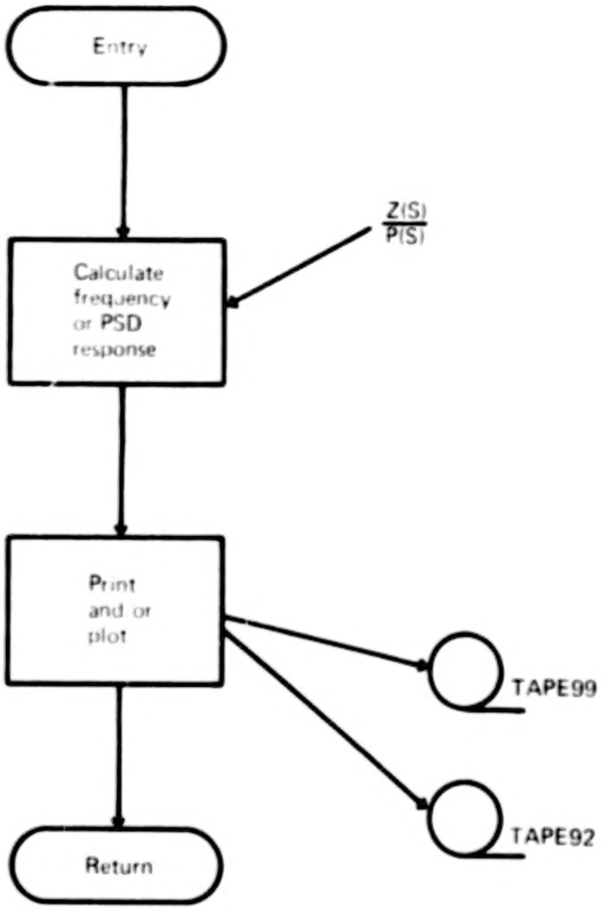


Figure 6.– Macro Flow Chart of Overlay (QRF.3.0)

Table 5. – Subroutines Called by Overlay (QRF,3,0)

Overlay (QRF,3,0)
<u>Program QRF30</u>

BDNYQ	(see tree on following page)	
CLEAR		
FPRINT	CLEAR	
	KGRID	DXDYV
		GRIDIV
		SETMIV
	KPLOT	POINTV
	PLTLT	IXV
		IYV
		LINEV
	PRINTV	
	RWI5	
	SCALE2	
	VPRNTD	A10R1
		BUFFIL
	WARN	
	WRTETP†	
GMARGN	RESP	
MNSQVS	(same list as FPRINT above)	
PEAKS	RESP	
PMARGN	RESP	
RESP		

† DYLOFLEX library routine

Table 5. – (continued)

BDNYQ	Calls		
ARCPLT	PLTLI	IXV	
		IYV	
	POINTV	LINEV	
	POLAR		
	POLER		
BRITEV			
CIRC1V	CIRC1V	PLTLI	IXV
			IYV
		XSCALV	
		YSCALV	
	BRITEV		
	SXDYV		
	FAINTV		
	FRAMEV		
	LABARC	LABLV	BEDEXP
			PRINTV
		NXV	
		NYV	
	RADI1V	IXV	
		IYV	
		LABLV (see above)	
		NXV	
		NYV	
		PLTLI (see above)	
	XSCALV		
	YSCALV		

† DYLOFLEX library routine

Table 5. – (concluded)

	Calls
BDNYQ	
CLEAR	
DRWCRD	LABLXC
	NODEC
	PLTLI (see above)
FAINTV	
KGRID	DXDYV
	GRID1V
	SETMIV
POINTV	
PRINTV	
SETMIV	
VPRNTD	A10R1
	BUFFIL
WARN	
WRTETP†	

† DYLOFLEX library routine

2.1.6 OVERLAY QRR,4,0

This overlay generates root locus plots given ZEROS(s), POLES(s), root locus eigenvalues, and plot specifications. All of this input information is read from data cards or calculated in the QRM,0,0 overlay and passed into QRR,4,0 by named common.

Root locus eigenvalues are stored on the disk file TAPE96 and are read into blank common as soon as the overlay is entered. Once all of the root locus points have been read, each of the complex arrays corresponding to ZEROS(s), POLES(s), and root locus eigenvalues are sorted into descending order using the imaginary part of the arrays as the sort criteria. The arrays are then plotted in accordance to specifications established by the user.

If QR is performing a COMPUTE FLUTTER AND ROOT LOCUS analysis during execution of this overlay, the ZEROS(s) and POLES(s) complex arrays are written to disk file TAPE96 for eventual processing in overlay QRR,5,0.

Figure 7 is a macro flow chart for the QRR,4,0 overlay.

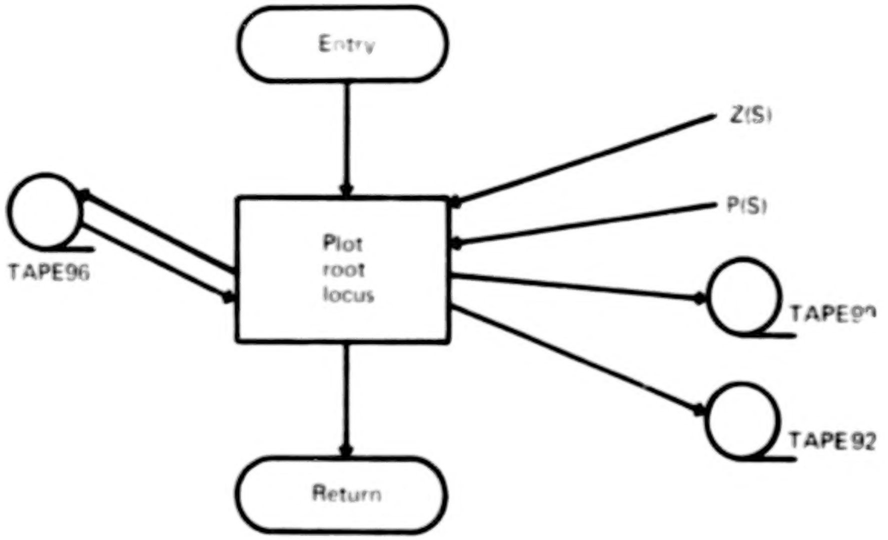


Figure 7. – Macro Flow Chart of Overlay (QRR,4,0)

Table 6 displays a map of subroutines called in QRR,4,0.

2.1.7 OVERLAY QRR,5,0

This overlay generates V-g plots given eigenvalues, k-values, reduced frequency reference length, and plot specifications. All of this input information is read from data cards or calculated in the QRM,0,0 overlay and passed into QRR,5,0 by named common.

Table 6. – Subroutines Called by Overlay (QRR,4,0)

Overlay (QRR,4,0) Program QRR40	
CLEAR	
FSF†	
KGRID	DXDYV
	GRID1V
	SETMIV
KPLOT	POINTV
PRINTV	
PRPLOT	BLKFLL
	PRLINE
SORT	
VPRNTD	A10R1
	BUFFIL
WARN	
WRTETP†	

† DYLOFLEX library routine

Eigenvalues are stored on the disk file TAPE96 and are read into blank common as soon as the overlay is entered. As each set of eigenvalues is read, arrays corresponding to velocity (knots or kilometers) and damping are calculated. Once all of the eigenvalues have been read, the arrays of velocity, damping, and cycles per second are sorted into ascending order using velocity as the sort criteria. The arrays are then plotted in accordance to specifications established by the user.

Figure 8 is a macro flow chart for the QRR,5,0 overlay.

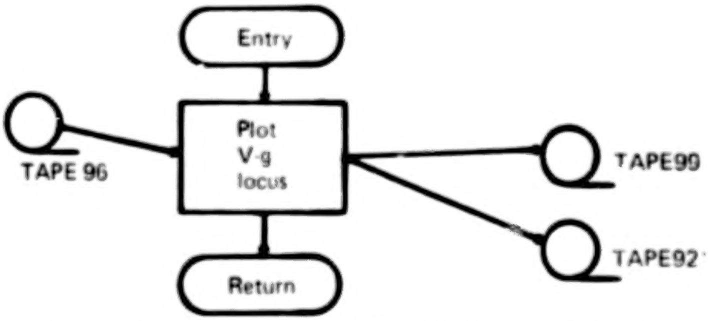


Figure 8. – Macro Flow Chart of Overlay (QRR,5,0)

Table 7 displays a map of subroutines called in QRR.5,0.

Table 7. – Subroutines Called by Overlay (QRR,5,0)

Overlay (QRR, 5,0) Program QRR50	
CLEAR	
FSF†	
KGRID	DXDYV
	GRID1V
	SETMIV
PRINTV	
PRPLOT	BLKFLL
	PRLINE
STRMOV†	
VPRNTD	A10R1
	BUFFIL
WARN	
WRTETP†	

† DYLOFLEX library routine

2.1.8 OVERLAY QRU.5,0

This overlay (see table 8) reads data associated with the UNSTEADY AIRFORCE COEFFICIENTS data card. Section 4.9 of volume I explains in detail the unsteady aerodynamic options in QR.

Table 8. – Subroutines Called by Overlay (QRU,5,0)

Overlay (QRU,5,0) Program QRU50
RWE12
RWI5

2.1.9 OVERLAY QRU,6,0

The purpose of this overlay (see table 9) is to set up calls to overlays QRU,6,1; QRU,6,2; QRU,6,3; and QRU,6,6.

Table 9. – Subroutines Called by Overlay (QRU,6,0)

Overlay (QRU,6,0) Program QRU60	
AUTOFL	REQFL†
FSR†	

† DYLOFLEX library routine

2.1.10 OVERLAY QRU,6,1

The purpose of this overlay (see table 10) is to read data cards associated with the COMPUTE UNSTEADY FREQUENCY RESPONSE data card and increment the various loop counters associated with the unsteady frequency response option. Section 4.9 of volume I explains the detail of this QR option.

Table 10. – Subroutines Called by Overlay (QRU,6,1)

Overlay (QRU,6,1,) Program QRU61
ATMS62
ENDQR
RWI5

2.1.11 OVERLAY QRU,6,2

The purpose of this overlay is to calculate the lower and upper frequency limits of the unsteady frequency response and to establish an array of k- values that can be used during the unsteady frequency response.

Using the first aerodynamic influence coefficient matrix, its k-value, and equation 4.9-1 from volume I, a set of eigenvalues can be calculated that identify the range of frequencies associated with an aircraft. The eigenvalue array is scanned for the minimum and maximum non-zero roots. The lower and upper frequency bounds are set one decade lower and higher from the minimum and maximum complex roots.

Once the frequency limits have been calculated, QR reads the specified disk containing aerodynamic influence coefficient matrices for the k-value associated with each of these matrices. The k-values are stored in an array for use in overlay QRU,6,3.

Figure 9 is a macro flow chart for the QRU,6,2 overlay.

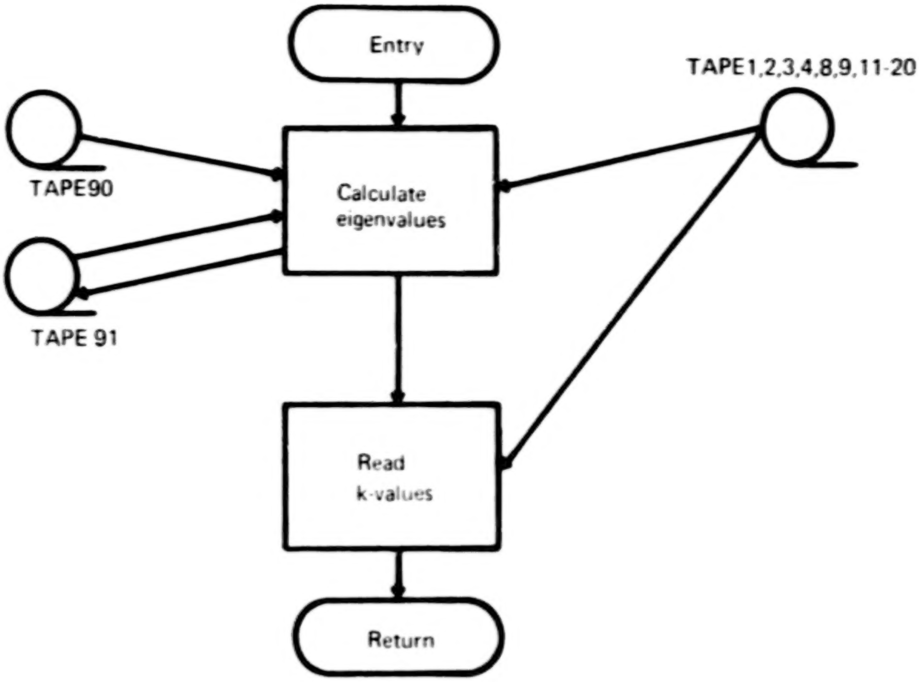


Figure 9.– Macro Flow Chart of Overlay(QRU,6,2)

Table 11 displays a map of subroutines called in QRU,6,2.

2.1.12 OVERLAY QRU,6,3

The purpose of this overlay is to calculate the unsteady frequency response of equation 4.9-1 in volume I. The user must provide the force vector array **F** and the normal coordinate position **q** as well as a number of other related inputs. Section 4.9 of volume I explains the detail of these inputs. Much of the input is read from data cards and stored in named common before this overlay is called.

The frequency response itself is calculated by substituting j into equation 4.9-1 and solving the resultant set of complex simultaneous equations for the user-identified normal coordinate position. Additional values of omega are picked in an ascending order such that the db change and phase change are less than the user-specified limit. As omega ascends in value, it is necessary to calculate the k-value corresponding to it. If the calculated k-value is greater than the k-values associated with the two aerodynamic influence coefficient matrices residing in core, new matrices are read from disk until an interpolation for the k-value calculated from omega can be made.

The user may use disk numbers 1,2,3,4,8,9,11-20 to input the aerodynamic influence coefficient matrices.

Table 11. – Subroutine Called by Overlay (QRU,6,2)

Overlay (QRU,6,2) Program QRU62		
AUTOFL	REQFL†	
BSF*		
EPLUSD		
MOMEGA		
ORDER		
QRROOT (see following page)		
RDMAT	READTP	
	TAPDIM	ENDQR
		WARN
RPRINT	SECOND*	
	STRMOV†	
RSTORE	AUTOFL	RECFL†
WARN		

† DYLOFLEX library routine
* CDC FORTRAN library routine

Table 11. – (continued)

Routines called by QRROOT

AUTOFL	REQFL†		
CHESS	CLEAR		
	WRITUM	WRITER	
COREIG			
CSAVE2			
EIGMAT	EXPAND	CLEAR	
		ROTSM	
		WRITUM	WRITER
	HELPER	CLEAR	
		COFSIN	
		COLCHK	
		ELMCHK	CLIM
		POLCHK	
		REGRUP	
		ROOTS	MULLP*
		WRITUM	WRITER
	NORMAL	CLEAR	
		CLIM	
		FINDPV	
		GAUSS	
		JJFIND	
		NEGMAT	
		REGRUP	
		SWAP	
		WRITUM	WRITER
GAUSPM	AUTOFL	REQFL†	
PRECON	EQVCOL		
	IRSCAL		
RHESS			
RQREIG			

† DYLOFLEX library routine
* CDC FORTRAN library routine

Table 11. – (concluded)

RSTOR2	AUTOFL	REQFL†	
SECOND*			
SVFORM	EXPAND (see above)		
	NORMAL (see above)		
	MATPRT	WA10	
	SETORD	AUTOFL	REQFL†
		JJFIND	
		SWAP	
WRITUM	WRITER		

† DYLOFLEX library routine
* CDC FORTRAN library routine

Figure 10 is a macro flow chart for the QRU,6,3 overlay.

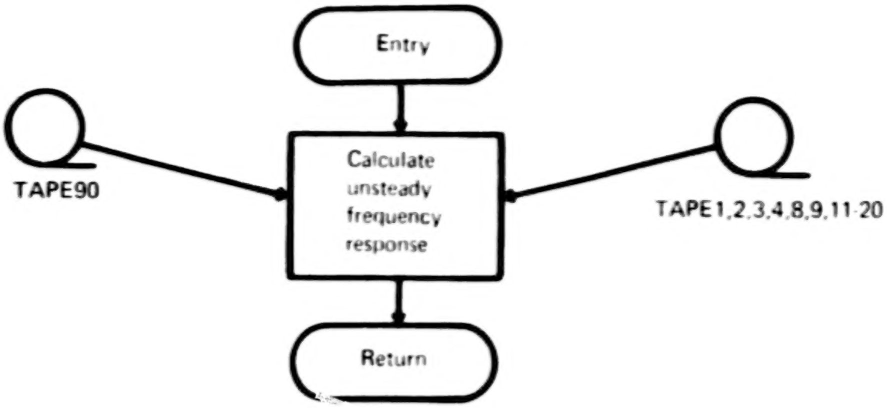


Figure 10. – Macro Flow Chart of Overlay (QRU,6,3)

Table 12 displays a map of subroutines called in QRU,6,3.

Table 12. – Subroutines Called by Overlay (QRU,6,3)

Overlay (QRU,6,3)
Program QRU63

AUTOFL	REQFL†		
IORESP	BSF*		
	CGLESM†		
	CLEAR		
	ETIMEC		
	RDMAT	READTP†	
		TAPDIM	ENDQR
			WARN
	SUBJOM		
MOVEFR			
RDMAT (see above)			
RSTORE	AUTOFL	REQFL†	
TRANS	CLEAR		
	ENDQR		

† DYLOFLEX library routine
* CDC FORTRAN library routine

2.1.13 OVERLAY QRU.6.6

The purpose of this overlay is to calculate the maximums and minimums, gain, and phase margin and to print plot the frequency response calculated in overlay QRU.6.3. Input data for this overlay are passed by named common from the QRM.0.0 overlay. The frequency response is passed from the QRU.6.3 overlay in blank common.

Figure 11 is a macro flow chart for the QRU.6.6 overlay.

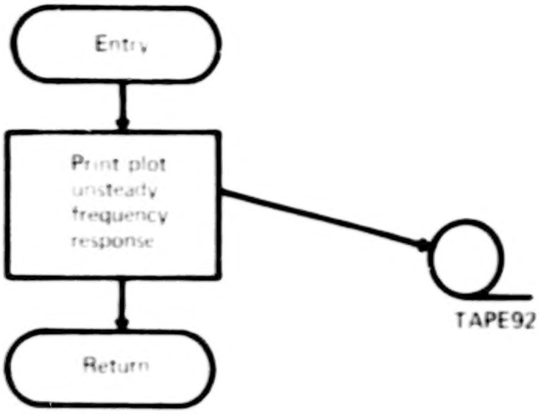


Figure 11. — Macro Flow Chart of Overlay (QRU,6,6)

Table 13 displays a map of subroutines called in QRU.6.6.

Table 13. — Subroutines Called by Overlay (QRU,6,6)

Overlay (QRU,6,6) Program QRU66	
BDNYQ (see table 5)	
UGMARG	TBLU1†
UNSQVS (see calls to MNSQVS in table 5)	
UPEAKS	
UPMARG	TBLU1†

† DYLOFLEX library routine

2.1.14 OVERLAY QRU.7.0

The purpose of this overlay is to set up calls to overlays QRU.7.1; QRU.7.3; and QRU.7.5.

2.1.15 OVERLAY QRU,7,1

The purpose of this overlay is to read data cards associated with the COMPUTE FLUTTER 2 data card, increment various loop counters associated with the option, and read the disk containing aerodynamic influence coefficient matrices for the k-value associated with each of these matrices. The k-value is stored in an array for use in overlay QRU,7,3.

Figure 12 is a macro flow chart for the QRU,7,1 overlay.

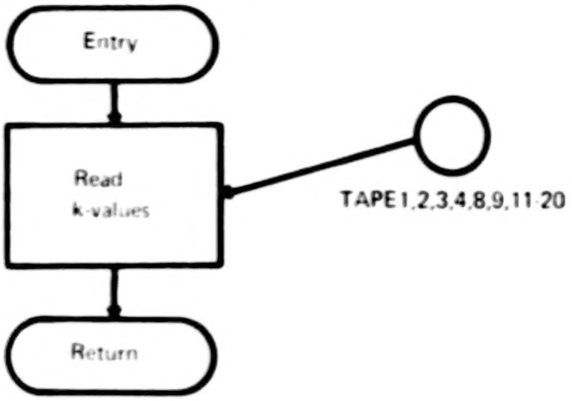


Figure 12. – Macro Flow Chart of Overlay (QRU,7,1)

2.1.16 OVERLAY QRU,7,3

The purpose of this overlay is to calculate the eigenvalues of the matrix polynomial given by equation 4.9-1 in volume I. Eigenvalues are calculated for each k-value that is input by the user.

Overlay QRU,7,1 establishes a table of k-values that are used for interpolation purposes in this overlay. QR positions the disk file containing the aerodynamic influence coefficient matrices and reads two of the matrices into central memory. The first user specified k-value is inspected and if it lies between the first two k-values that reside on disk, the eigenvalues are calculated. Otherwise, another matrix is read from disk. This procedure is repeated until eigied a k-value that is below or above the range of the table read in overlay QRU,7,1, a diagnostic is written onto the output file. In addition, the user must input k-values for flutter analysis in ascending order.

Figure 13 is a macro flow chart for the QRU,7,3 overlay.

Table 14 displays a map of subroutines calls in QRU,7,3.

2.1.17 OVERLAY QRU,7,5

The purpose of this overlay is to plot the V-g analysis produced by overlay QRU,7,3.

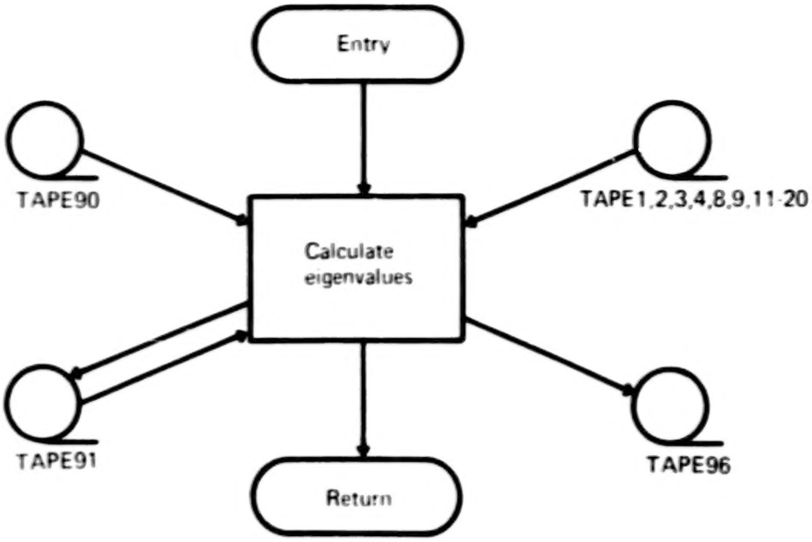


Figure 13. – Macro Flow Chart of Overlay (QRU,7,3)

Table 14. – Subroutines Called by Overlay (QRU,7,3)

Overlay (QRU,7,3) Program QRU73			
AUTOFL	REQFL†		
FLUTEI	AUTOFL	REQFL†	
	BSF*		
	ENDQR		
	FORMPM		
	ORDER		
	QRROOT (see table 11)		
	RDMAT	READTP†	
		TAPDIM	ENDQR
			WARN
	RPRINT	SECOND*	
		STRMOV†	
	RSTORE	AUTOFL	REQFL†
	WRITUM	WRITER	

† DYLOFLEX library routine
* CDC FORTRAN library routine

Eigenvalues are read into blank common from disk file TAPE96 as soon as the overlay is entered. As each set of eigenvalues is read, arrays corresponding to velocity (knots or kilometers) and damping are calculated. Once all of the eigenvalues have been read, the arrays of velocity, damping, and cycles per second are sorted into ascending order using velocity as the sort criteria. The arrays are then plotted in accordance to specifications established by the user.

Figure 14 is a macro flow chart for the QRU,7,5 overlay.

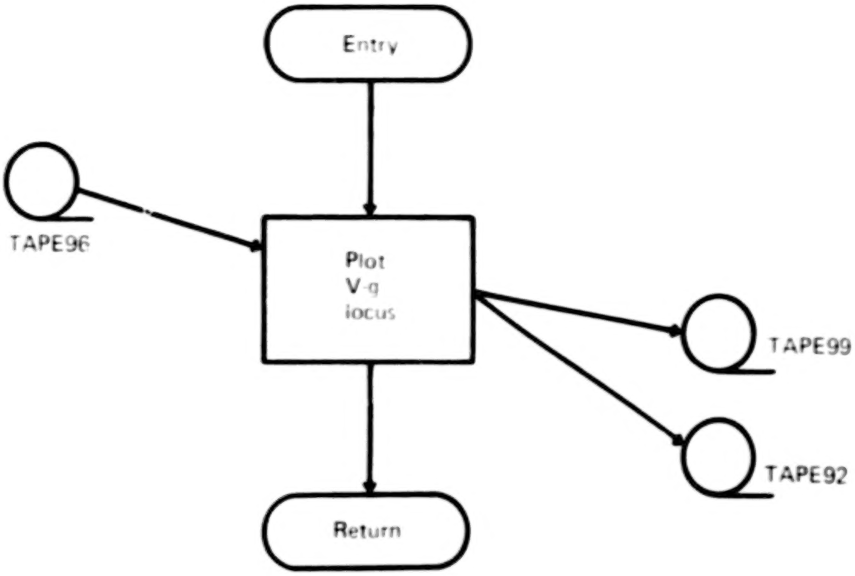


Figure 14. – Macro Flow Chart of Overlay (QRU,7,5)

Table 15 displays a map of subroutine calls in QRU,7,5.

2.2 DATA BASES

The data bases for QR include several input, output, and scratch files as well as a large number of FORTRAN common blocks.

2.2.1 INPUT DATA

Data is input to the program in three forms; data cards, magnetic files, and user-provided FORTRAN subroutines.

2.2.1.1 Card Input Data

For a complete description of card input data, see section 6.3 of the QR usage document, volume I.

Table 15. – Subroutines Called by Overlay (QRU,7,5)

Overlay (QRU,7,5) Program QRU75	
CLEAR	
KGRID	DXDYV
	GRID1V
	SETMIV
PRINTV	
PRPLOT	BLKFIL
	PRLINE
STRMOV†	
VPRNTD	A10R1
	BUFFIL
WARN	
WRTETP†	

† DYLOFLEX library routine

2.2.1.2 Magnetic File Input Data

Magnetic tape can be used to supply matrices required by QR. These files should be copied to disk before being used with the QR program. Permissible disk file names are TAPE1, TAPE2, TAPE3, TAPE4, TAPE8, TAPE9, TAPE11, TAPE12, TAPE13, TAPE14, TAPE15, TAPE16, TAPE17, TAPE18, TAPE19, TAPE20. Sections 6.1 and 6.3 of the QR User's Manual provide additional information about tape/disk input.

2.2.1.3 FORTRAN Input

The elements of the square polynomial matrix and column vector array used by QR can be input through the use of user-supplied FORTRAN subroutines. Sec. 6.8 of the QR User's Manual describes this input option.

2.2.2 OUTPUT DATA

The results of a program execution can be printed or written on magnetic tape. In addition, a plot tape (TAPE99), plot vector tape, and punched matrix cards can be generated by the program.

2.2.2.1 Printed Output Data

A short sample of QR printed output is included in section 4.9 of volume I.

2.2.2.2 Punched Card Output Data

The PUNCH option described in section 6.3.2 of volume I allows the user to punch the square polynomial matrix used by QR with the format 4(3I2,E14.7). The matrix punched on cards may be read later by QR using the MATRIX or REPLACE card input options.

2.2.2.3 Magnetic File Output Data

Two magnetic disk files are often produced by QR executions. TAPE92 contains data records that represent data appearing on SC4020 plots.

TAPE99 is a binary file of plot instructions and data points for the SC4020 plotter. Users normally pass this information to the SC4020 plotter through use of the PLOTFIL command. Reference 3 provides additional information on this command.

2.2.3 INTERNAL DATA

Two methods are used to pass data from one portion of the program to another - common blocks and temporary disk storage files.

2.2.3.1 Common Blocks

LABELED common blocks are used for communication between subroutines and overlays. The block names and contents are described on the following pages. Table 16 shows the overlays in which each block is defined.

BLANK common is used in secondary overlays as a variable length working area.

Table 16. – Labeled Common Blocks Used in Each Overlay

COMMON BLOCK NAME	QRM,0,0	QRM,1,0	QRM,7,0	QRT,2,0	QRF,3,0	QRR,4,0	QRR,5,0	QRU,5,0	QRU,6,0	QRU,6,1	QRU,6,2	QRU,6,3	QRU,6,6	QRU,7,0	QRU,7,1	QRU,7,3	QRU,7,5
ALPHT	x			x	x	x	x						x				x
ARRAYK									x		x	x		x	x	x	
BATCH	x	x	x	x	x	x	x		x				x				x
COUNT	x		x	x	x	x	x						x				x
CPOLE	x		x	x													
CZERO	x		x	x													
DATRFV	x	x															
DCGAIN	x																
DEF	x			x													
DUMBS	x		x	x	x	x	x						x				x
DYN	x										x	x			x	x	
FCONST														x	x	x	
FFX	x			x													
FFY	x			x													
FMARG	x			x	x	x	x						x				x
FMTLBL	x																
FRMN	x			x													
FRQRES	x				x								x				
GAIN	x																
GUSTS	x				x								x				
IBLANK	x			x		x	x										x
ICARD	x	x															
ICOMTY	x	x		x	x	x	x	x	x	x			x	x	x		x
IFORM	x															x	
IGARFR	x					x	x							x		x	x
ILOW	x																
IMASK	x			x	x	x	x						x				x
IPA	x	x															
ITEM	x	x	x														
ITIME1	x			x													
JKL	x																
LWA	x								x					x		x	
MATCOM	x										x						
MAXFL	x		x														
MAXFR	x																
MCMPLX	x				x						x			x			
MNO	x			x	x												
MSZ	x											x					
NBS	x																
NFF	x			x													
NODAIG	x																
NUKNOW									x	x		x					

Table 16. – (concluded)

COMMON BLOCK NAME	QRM,0,0	QRM,1,0	QRM,7,0	QRT,2,0	QRF,3,0	QRR,4,0	QRR,5,0	QRU,5,0	QRU,6,0	QRU,6,1	QRU,6,2	QRU,6,3	QRU,6,6	QRU,7,0	QRU,7,1	QRU,7,3	QRU,7,5
NUSFRP									x	x		x	x				
OMEGAS									x		x	x					
OOS												x					
OUTI	x			x	x	x	x						x				x
PHASE	x																
PLOT	x					x	x									x	x
PLOTFL	x						x										x
PLOTFR	x				x								x				
POLE	x					x											
PPLOY	x																
PSDPLT	x				x								x				
PTITLE	x			x	x	x	x						x				x
PTRPLT																	x
RCM	x											x					
ROWCOL	x																
RWBLCK	x																
RWBUFF	x	x		x	x	x	x				x	x	x		x	x	x
SCLXX	x				x								x				
SCLYY	x				x								x				
TELEX	x	x		x	x	x	x			x							x
TIMES	x			x													
TIMOLD	x																
TITL	x			x	x	x	x						x				
TPJUNK	x			x					x	x	x	x	x	x	x	x	x
TPJNK					x	x	x										
UCONST									x	x	x	x					
UNITS	x						x	x		x					x		x
UNSTDY	x							x	x	x	x	x		x	x	x	
UPOINT									x	x	x	x	x				
USERK														x	x	x	
VECTOR	x											x					
VTS	x			x		x	x										x
ZERO	x					x											
ZPOLY	x																
BLANK	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

LABELED COMMON NAME: ALPHT

DESCRIPTION: Table of SC4020 hollerith characters

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IALPHA	I	47		SC4020 hollerith table

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ARRAYK

DESCRIPTION: Array of K-Values used by QR

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ARRAYK	R	30		Array of k-values read from disk

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: BATCH

DESCRIPTION: Logical flag

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	BATCH	L	1		This flag is .TRUE. if batch execution, .FALSE. otherwise

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME COUNT

DESCRIPTION Block of counters for SC4020 buffer

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NWORD	I	1		Number of words in SC4020 buffer
2	NCHR	I	1		Number of characters per CDC word
3	IWORD	I	1		Word to which program is positioned in SC4020 buffer
4	ICHR	I	1		Character to which program is positioned in IWORD

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME CPOLE

DESCRIPTION Array of noncancelled poles

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NCPOLE	I	1		Number of noncancelled poles
2	CPOLE	C	60		Eigenvalues that are designated POLES.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME CZERO

DESCRIPTION Array of noncancelled zeros.

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NCZERO	I	1		Number of noncancelled zeros
2	CZERO	C	60	ZERO(S)	Eigenvalues that are designated ZEROS

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME DATREV					
DESCRIPTION Contains date of execution					
Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	RDT	I	1		Date of execution

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME DCGAIN

DESCRIPTION DC gain of feedback control system, etc

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	DCGAIN	R	1		DC gain of feedback control system
2	NINTEG	I	1		Number of poles at origin user wishes to add to feedback control system
3	NDIFFER	I	1		Number of zeros at origin user wishes to add to feedback control system
4	ISTORE	I	1		Variable flag QR uses to designate POLES, ZEROS, ROOT LOCUS, etc

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME DEF

DESCRIPTION Cancellation tolerances

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	XREAL	R	1		Maximum real difference allowed for cancellation of poles and zeros
2	XIMAG	R	1		Maximum imaginary difference allowed for cancellation of poles and zeros
3	CRATIO	R	1		Maximum ratio allowed for cancellation of poles and zeros Section 4.7.4 of volume I

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME DUMBS

DESCRIPTION Array of SC4020 instructions

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ICHAR	I	6		Array of SC4020 instructions ready for transfer into SC4020 buffer

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME DYN

DESCRIPTION Block of matrix dimensions and pointers

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MATORD	I	1		Number of rows in A matrix
2	MATDEI	I	1		Highest power of S plus 1
3	NGRCOL	I	1		Start of COL array in RWBUFF
4	NGRGRW	I	1		Start of GROW array in RWBUFF
5	NGRGCL	I	1		Start of GCOL array in RWBUGG
6	NGRIPV	I	1		Start of IPIV array in RWBUFF
7	NGRNGR	I	1		Two times number of guessed roots
8	NGRID	I	1		Start of ID array in RWBUFF
9	NGROOT	I	1		Start of ROOT array in RWBUFF

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME FCONST

DESCRIPTION Block of constants for COMPLETE FLUTTER 2

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IALT	I	1		Counter of altitude positions
2	NK	I	1		Number of k-values
3	NOEFK	I	1		Number of k-value matrices to read
4	MN	I	1		k-value matrices

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME FFX

DESCRIPTION Time values for forcing function

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	FFX	R	12		Time values that make up forcing function. F$_X$(t)

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: FFY

DESCRIPTION: Forcing function values

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	FFY	R	12	$F_2(t)$	Values of forcing function

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: FMARG

DESCRIPTION: Raster positions of grid

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	FMTL	R	1		MTL
2	FMTR	R	1		1023-MTR
3	FMTB	R	1		MTB
4	FMTT	R	1		1023-MTT
5	ML	I	1		Left of grid is ML rasters left of screen edge
6	MR	I	1		Right of grid is MR rasters left of screen edge
7	MB	I	1		Bottom of grid is MB rasters up from screen edge
8	MT	I	1		Top of grid is MT rasters down from screen edge

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: FMTLBL

DESCRIPTION: Format of SC4020 label values

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IFMT(2)	I	2		Format of SC4020 label value
2	LMARY	I	1		
3	LADJX	I	1		Constants for label positioning. See subroutine NODEC
4	LBMARX	I	1		

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME FRMN

DESCRIPTION Frame count

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NFRM	I	1		Frame count

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME FRQRES

DESCRIPTION Calculation and print indicators for frequency response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IFRQ1	I	1		Controls if frequency response is done during root locus
2	IFRQ2	I	1		Controls print detail of frequency response
					See section 4.7.6 of volume I for additional detail.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME GAIN

DESCRIPTION Gains of feedback control system

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NGAIN	I	1		Number of gains to be evaluated in root locus sequence
2	GAIN	R	48		Gain values

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME GUSTS

DESCRIPTION Coefficients for gust spectrum

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IGUST	I	1		Flag to determine if PSD is to be calculated
2	AGUST	R	1		
3	BGUST	R	1		
4	CGUST	R	1		Coefficients for gust spectrum used in PSD response. See section 4.7.6 of volume I for additional detail
5	DGUST	R	1		
6	PSTART	R	1		

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME IBLANK

DESCRIPTION Hollerith blank

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IBLANK	H	1		Hollerith blank

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ICARD

DESCRIPTION Temporary buffer for QR data card

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ICARD	H	8		Temporary buffer for QR data card

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ICOMTY

DESCRIPTION Indicator block for QR input commands

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ICOMTY	I	1		Error flag for interactive QR; set 0 during batch
2	ICMD	I	1		Integer value of current QR command

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME IFORM

DESCRIPTION Indicator for printing polynomials formed in QR

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IFORM	L	1		If IFORM is .TRUE., polynomial coefficients are always printed by QR.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME IGARFR

DESCRIPTION Block of variables used in unsteady aerodynamics options

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	GINVRS	R	6	$1/\gamma$	Inverse gamma values
2	ALTITU	R	6		Altitude positions for flutter analysis
3	RFRL	R	1	b	Reduced frequency reference length
4	NIGFAA	I	1		Number of inverse gamma values
5	NIG	I	1		Current inverse gamma value subscript
6	NKVALU	I	1		Number of k-values on disk
7	VALUEK	R	1		Current k-value
8	NK	I	1		k-value counter

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ILOW

DESCRIPTION Hollerith table of QR commands

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ILOW	H	74		Table of QR commands, long form
2	ILOWA	H	74		Table of QR commands, short form

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME IMASK

DESCRIPTION Masking variables for SC4020 plotting

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MASK	I	6		Masking variables for SC4020 plotting

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: IPA

DESCRIPTION: Buffer area for FETEDIT files

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	TAPBUFF	I	66		Buffer area
2	FET	I	340		20 FET areas for pooled buffers

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ITEM

DESCRIPTION CRU cost tracking information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ITEM	I	16		Array that is passed to subroutine COST.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ITIME1

DESCRIPTION Block of print and plots indicators for time response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ITIME1	I	1		Indicators for printing and plotting time response. See section 4.7.5 of volume I
2	ITIME2	I	1		
3	ITIME3	I	1		
4	ITIME4	I	1		
5	YQTOP	R	2		Optional maximum and minimum ordinate values for time response

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME JKL

DESCRIPTION Leading coefficients of POLE and ZERO polynomials

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	PCOEF	C	1		Leading coefficient of POLE polynomial
2	ZCOEF	C	1		Leading coefficient of ZERO polynomial

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME LWA

DESCRIPTION Last word address of QRM. 0.0 overlay

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	LWA	I	1		Last word of QRM. 0.0 overlay Note, however, that LWA is adjusted during QRU overlay executions to include arrays and machine code that cannot be destroyed

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MATCOM

DESCRIPTION Constants used for eigenvalue rooting

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
2	COFF	C	1		Leading coefficient of A matrix
2	PIVOT	C	1		Value of last pivoted point referred to
3	MORIG	I	1		Original number of rows in A matrix
4	NORM	L	1		NORM is .TRUE. if subroutine CLIM is to do Gaussian elimination
5	SINGLR	L	1		SINGLR is .TRUE. if A is sigular
6	VTEST	L	1		VTEST is .TRUE. if state variable expansion is underway
7	ZTEST	R	1		A number of considered to be zero if less than or equal ZTEST
8	EPS	R	1		Two polynomials are equal if their normalized coefficients vary by no more then EPS

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MATCOM (continued)

DESCRIPTION

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
9	MVSHIF	I	1		A shift integer used to locate the (T) or (V) matrix
10	MVSTOP	I	1		A limit for the rows of (T) or columns of (V)
11	DIVCHK	R	1		Number less than DIVCHK is considered zero
12	TRACE	L	1		TRACE is .TRUE. if a detail of eigenvalue rooting is desired
13	NGR	I	1		Number of guessed roots
14	NROOT	I	1		Number of eigenvalues
15	NRADD	I	1		Number of extracted eigenvalues

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MAXFL

DESCRIPTION Maximum field length used by QR

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MAXFL	I	1		Maximum field length used by QR

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MAXFR

DESCRIPTION Maximum frequency of roots printed by QR

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MAXFR	R	1		Eigenvalues having a complex absolute value greater than MAXFR will not be printed by QR during a KIT execution

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MCMPLX

DESCRIPTION Integer flag that determines if real or complex matrices

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MCMPLX	I	1		MCMPLX is 1 for real matrices and 2 for complex matrices

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MNO

DESCRIPTION Leading coefficient of polynomial ratio

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	FREQN	C	1		Leading coefficient of ZEROS (S)/POLES (S).

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME MSZ

DESCRIPTION Size of matrix stored on TAPE 90 and TAPE 91

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MSZ	I	1		Size of matrix stored on TAPE90 and TAPE91

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME NBS

DESCRIPTION Block of Body station information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NBS	I	1		Number of body stations
2	X	R	18		Body station positions

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: NFF

DESCRIPTION Number of forcing function points

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NFF	I	1		Number of forcing function points

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME NODIAG

DESCRIPTION Logical Flag for Printing Diagnostics

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NODIAG	L	1		If NODIAG is .TRUE., all diagnostics generated by QR will be printed.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: NUKNOW

DESCRIPTION: Block of data for unsteady frequency response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NUKNOW	I	1		Number of unknown in unsteady frequency response.

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME NUSFRP

DESCRIPTION Block of data for unsteady frequency response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NUSFRP	I			Number of unsteady frequency response points.
2	NRHSV	I			Number of right-hand vectors
3	NUPSD	I			Flag indicating if PSD response is to be done in addition to unsteady frequency response.

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME OMEGAS					
DESCRIPTION Values of beginning and ending omega values for unsteady frequency response					
Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	OMEGAB	R	1		Lowest omega value
2	OMEGAF	R	1		Largest omega value

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: OOS

DESCRIPTION: Values of omega and omega squared for unsteady frequency response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	OMEGA	R	1	ω	Omega
2	OMEGAS	R	1	ω^2	Omega^2

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME OUT1

DESCRIPTION SC4020 plot buffer

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IOUT	I	408		SC4020 plot buffer

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PHASE

DESCRIPTION Phases of feedback control system

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPHASE	I	1		Number of phases to be evaluated in root locus sequence
2	PHASES	R	6		Phase values in degrees

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PLOT

DESCRIPTION Root locus plot specifications

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPLOT	I	1		Number of root locus regions
2	XLEFT	R	6		Minimum values of real axis
3	XRT	R	6		Maximum values of real axis
4	YTOP	R	6		Maximum values of imaginary axis
5	YBOT	R	6		Minimum value of imaginary axis
6	NPHA	I	1		Flag indicating if SC4020 plots phase locus are to be separate
7	IRLPF	I			Flag indicating if SC4020 plots are to be produced
8	NRNTS	I			Number of loci sets on TAPE96
9	MFILE	I			Number of loci files on TAPE96
10	NRNTSS	I			Count of loci sets on TAPE96 for flutter locus

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PLOTFL

DESCRIPTION Plot specifications for V-g plots

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	GMIN	R	1		Minimum dumping value
2	GMAX	R	1		Maximum dumping value
3	VMIN	R	1		Minimum velocity
4	VMAX	R	1		Maximum velocity
5	CMIN	R	1		Maximum cps value
6	CMAX	R	1		Minimum cps value
7	NPLTFL	I	1		Flag indicating if there will be V-g plots

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PLOTFR

DESCRIPTION Plot specification for Bode and Nyquist Plots

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPLOTF	I	1		Number of regions to plot
2	IFROPT	I	1		Flag indicating automatic scaling
3	IOMGAL	I	6		Log to base 10 of left limit of frequency scale
4	IOMGAR	I	6		Log to base 10 of right limit of frequency scale
5	IBDNYQ	I	6		Flag indicating Bode and/or Nyquist plots
6	IDBSCL	I	6		Flag indicating automatic scaling
7	ILEFTW	I	6		Temporary location for left scale
8	IRIGTW	I	6		Temporary location for right scale

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME POLE

DESCRIPTION Array of user indicated POLES

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPOLE	I	1		Number of eigenvalues in POLE array
2	POLE	C	60		Eigenvalues that have been designated POLES(S)

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PPOLY

DESCRIPTION Polynomial coefficients of POLES(S) array

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPPOLY	I	1		Highest power of S associated with POLES(S) array
2	PPOLY	C	61		Polynomial coefficients of POLES(S) array

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PSDPLT					
DESCRIPTION PSD plot specifications					
Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPSPP	I	1		Flag indicating if there are to be PSD plots
2	ISCPS	I	1		Flag indicating cps or radians along abscissa
3	FLR	R	12		Maximums and minimums of abscissa
4	IORDS	H	8		Optional title for PSD plot

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PTITLE

DESCRIPTION Plot title information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	PTITLE	H	12		Three lines of title information for SC4020 plots

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME PTRPLT

DESCRIPTION Storage for COMPUTE FLUTTER 2

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	G	R	1250		Damping values for V-g plots
2	KNOTS	R	1250		Velocity values for V-g plots
3	CPS	R	1250		CPS value for V-g plots
4	SYMBOL	R	1250		Symbols for V-g plots

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

CONTENTS

LABELED COMMON NAME RCM

DESCRIPTION Gain location information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPLOC	I	1		Number of gain locations in A matrix
2	IRCM	I	24		Row, column, and power of S locations of gains

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ROWCOL

DESCRIPTION Logical flag that indicates rows or columns of the A matrix

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IRC	L	1		If IRC is .TRUE., scanning a row, .FALSE. means column.

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: RWBLCK

DESCRIPTION: Scratch area for input/output subroutines

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	RWBLCK		12		Scratch array for several input/output subroutines.

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME RWBUFF

DESCRIPTION Scratch area

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	RWBUFF		902		Scratch area for QR calculations

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: SCLXX

DESCRIPTION: Scaling for SC4020 plots

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	ASCAL	R	1		Slope of interpolation for X raster
2	BSCAL	R	1		Constant of interpolation for X raster

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME SCLYY

DESCRIPTION Scaling for SC4020 plots

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	CSCAL	R	1		Slope of interpolation for Y raster
2	DSCAL	R	1		Constant of interpolation for Y raster

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: TELEX

DESCRIPTION: Logical flag to indicate KIT

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	TELEX	L	1		TELEX will be .TRUE. if QR is executed from the terminal

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME TIMES

DESCRIPTION Block of information that defines time history

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	TIMEI	R	6		Starting time
2	TIMED	R	6		Time step increment
3	TIMEF	R	6		Ending time
4	NTIME	I	1		Number of time increments

* Refers to variable type   I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME TIMOLD

DESCRIPTION Cumulative CP time of eigenvalue rooting

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	TIMOLD	R	1		Cumulative CP time to eigenvalue rooting. At end of solution, CP required to calculate eigenvalues is calculated and printed

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME TITL

DESCRIPTION Title information for printer and SC4020 plots

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	TITLE	H	8		Title information for printer and SC4020 plots.

* Refers to variable type I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME TPJUNK and TPJNK

DESCRIPTION Scratch array for READTP and WRTETP subroutines

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NTAPE	I	1		Unit number
2	NFILE	I	1		Number of files to skip
3	NMAT	I	1		Number of matrices to skip
4	MATDG	I	1		Power of S matrix plus 1
5	NPRT	I	1		Print flag
6	NAME	I	1		Fixed point number designating matrix
7	M	I	1		Row dimension
8	N	I	1		Column dimension
9	B	I	12		Identification array
10	IRROR	I	1		Error return

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME UNCONST

DESCRIPTION Unsteady aerodynamics information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	CONST	R	1		Product of several constants and k-value
2	VALNEK	R	1		k-value
3	MN	I	1		Product of row and column dimension of airforce matrix

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME UPOINT

DESCRIPTION Unsteady aerodynamics information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	IALT	I	1		Altitude array dimension
2	NMATR	I	1		Number of matrices to forward space on first k-value
3	NK	I	1		Number of k-values
4	NCLOL	I	1		Flag that determines kind of frequency response
5	NFRLIM	I	1		Flag that determines detail of frequency response
6	NEOFK	I	1		Flag that determines if air force matrices have been normalized
7	NXYMI	I	1		Number of combinations for closed loop response analysis
8	IXYMI	I	1		Loop counter for NXYMI
9	IRHSV	I	1		Loop counter for number of vectors

* Refers to variable type
- I - Integer
- R - Real
- C - Complex
- L - Logical
- H - Hollerith

LABELED COMMON NAME: UNSTDY

DESCRIPTION: Unsteady aerodynamics information

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	MACHNO	R	1		Mach number
2	RFRL	R	1	b	Reduced frequency reference length
3	CF	R	1		Correction factor
4	RHO	R	1		Density of air
5	V	R	1		Free stream velocity
6	NALT	I	1		Number of altitudes
7	ALT	R	6		Altitude values
8	LRHOAV	L	1		If LRHOAV is .TRUE., RHO and V are to be calculated by ATMS62

* Refers to variable type: I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: USERK

DESCRIPTION: User-specified K-values

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	USERK	R	30		User-specified K-values

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME VECTOR

DESCRIPTION B matrix storage location

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	VECTOR	R	120		B matrix storage location

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: VTS

DESCRIPTION: Ordinate title information for this response

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	VTS	H	4		Ordinate title for time response

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME: ZERO

DESCRIPTION: Array of user indicated ZEROS

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NZERO	I	1		Number of eigenvalues in ZERO array
2	ZERO	L	60		Eigenvalues that have been designated ZERO(S)

* Refers to variable type:
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

LABELED COMMON NAME ZPOLY

DESCRIPTION Polynomial coefficients of ZERO(S) array

Number	Variable	T*	Dimension	Engineering Nomenclature	Description
1	NPZERO	I 1	1		Highest power of S associated with ZERO(S) array
2	ZPOLY	C			Polynomial coefficients of ZERO(S) array

* Refers to variable type
I = Integer
R = Real
C = Complex
L = Logical
H = Hollerith

2.2.3.2 Magnetic Files (Disk)

QR uses three disk files for temporary storage of data. Data is written and read from two of these files using BUFFER OUT and BUFFER IN respectively. The third file is written and read using unformatted FORTRAN WRITE and READ operations.

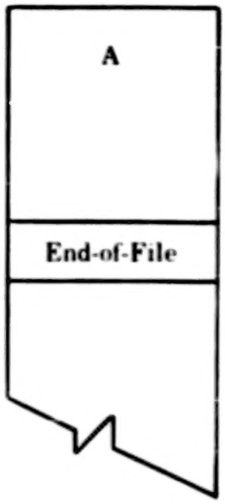
TAPE90 Scratch File

TAPE90 is a scratch file written and read by the QRM,0,0 overlay. This file contains an exact copy of the A matrix created by the user with the MATRIX/REPLACE/READ MATRIX TAPE etc. input data cards.

The file structure is:

Matrix Size | Matrix

NROWxNCOLxNDEGx2



where:

NROW = the number of rows

NCOL = the number of columns

NDEG = the highest power of S plus 1

designates complex elements

TAPE91 Scratch File

TAPE91 is a scratch file written and read by the QRM,0,0 overlay. This file contains an exact copy of the matrix for which eigenvalues are to be calculated. If the algorithm for eigenvalues fails because of a singular matrix, QR will reread TAPE91, set a trace flag, and reattempt the eigenvalue solution printing traceback information on the output file until the singularity condition reoccurs.

The matrix stored on TAPE91 has the same dimensions as the matrix stored on TAPE90. (However, element values may be different.)

TAPE96 Scratch File

Tape96 is a scratch file written by the QRM.0.0 and QRU.7.3 overlays and read by the QRR.4.0, QR.5.0, and QRU.7.5 overlays. In addition, after the QRR.4.0 overlay has read the intended set of root locus points, it writes the open loop zeros and poles (from the root locus calculation) and an end-of-file onto TAPE96.

The file structure is:

Word 1	The number of eigenvalues (NR1)
Word 2	k-value associated with eigenvalues
Word 3	starting in word3
• • •	Eigenvalues in complex format. Two times NR1 words
Word 2*NR1 + 3	The number of eigenvalues (NR2)
Word 2*NR1 + 4	k-value associated with eigenvalues following
• • • •	Eigenvalues in complex format. Two times NR2 words

The file structure of – number of eigenvalues, k-value, eigenvalues – is repeated until all flutter eigenvalue rootings have been completed. If the program is performing a COMPUTE FLUTTER AND ROOT LOCUS operation, the open loop poles and zeros are written onto TAPE96 followed by an end-of-file once the gain and/or phase loci have been read by the root locus plotting overlay (QRR.4.0). The program then advances to the next k-value and writes a second set of records onto TAPE96 followed by another end-of-file. This process is repeated until all k-value rootings have been completed.

In the case of a COMPUTE FLUTTER, COMPUTE FLUTTER 2 or COMPUTE ROOT LOCUS, only one file is written.

2.3 SUBROUTINES

Short descriptions of the purpose of each function and subroutine used in QR follow in alphabetical order.

Subroutine AA (OMEGA, CPS, FMAG, DB, PHI)

AA sets up subroutine calls to print and plot frequency and PSD response, finds maximums and minimums, calculates gain margin, and phase margin for the unsteady frequency response option.

Subroutine ACCT

ACCT contains a number of entry points localizing calls to QR overlay segments. Entry points for ACCT are FRESP, FTPLOT, TRPLOT, ENDQR. Before an overlay is called, subroutine AUTOFL is called to set the core size for the overlay execution. Only the overlays that are called from more than one location in QR are centralized in this routine. The overlays called are: QRM,1,0; QRF,3,0; QRR,4,0; QRR,5,0; and QRM,7,0.

Subroutine ARCPLT (RAD, THETA, NUM, RI, AL, INC, IND, ISYM)

ARCPLT is a routine to connect points on a polar grid by linear lines in the polar space. It also may place a symbol at the point and may connect the points with linear lines in the cartesian space.

INPUT:

RAD	An array of radii in radial units
	An array of angles in degrees.
	RAD and THETA have 1 to 1 correspondence.
NUM	The number of points to process.
	≤ 1 no processing will take place.
AL	Angular offset in degrees. The value in the 3 o'clock position on the polar grid.
INC	Line density.
	< 1 Straight lines will connect the points.
	≥ 1 ARCS will be constructed between the points using chains of straight lines of INC angular length.
IND	Place a symbol on the point.
	≤ 0 Do not place a symbol.
	> 0 Do place a symbol.
ISYM	The number of the symbol to use.

OUTPUT: None.

Function ARCTAN (FRR,FRI)

ARCTAN calculates the phase angle in degrees associated with the complex number FRR + j FRI. The angle returned by ARCTAN will be in the range -180 to + 180 degrees.

Subroutine ARCIV (R, DR, M, LA, KDO)

ARCIV was designed to be a primary level routine in the creation of a polar grid. Care has been taken to insure uniform density lines and to provide for minimum execution time.

ARCIV draws the axes of equal radii for the polar grid.

INPUT

R	The radial dimension of the circle.
DR	The radial dimension increment to be used in forming the circles.
M	The index to be used with DR to form the circles. Every Mth DR will be formed.
LA	Specifies the major axis of the grid. = 1 The vertical axis is major. ≠ 1 The horizontal axis is major.
KDO	The nature of the display. = 0 Only outermost circle will be drawn. ≠ 0 Every Mth DR will be drawn along with the circle at R.

OUTPUT: None

Subroutine ASSIST

ASSIST reads namelist data from the INPUT file and writes each card image read to the OUTPUT file and disk unit 7 (TAPE7).This subroutine is described in more detail in section 6.8 of volume I.

Subroutine ATMS62 (ALT,SONIC,PRESS,DENSI,TEMP,MACHNO)

ATMS62 calculates atmospheric properties of the US standard atmosphere, 1962.

Subroutine AUTOFL

AUTOFL calculates the field length required for execution and requests a field length change through subroutine REQFL. In addition, the maximum field length and last word address are calculated. These variables will be written out if the TRACE flag is ".TRUE.".

Subroutine AXISD (IX, IY, IIX, 0) or (IX, IY, IIY, 1)

AXISD draws a horizontal or vertical line from point (IX,IY) in raster coordinates to (IIX,IY) or (IX,IIY). Note that (IIX-IX) ≥ 64 or (IIY-IY) ≥ 64.

Subroutine A10R1 (A,B,N)

A10R1 takes N hollerith characters stored in array A and creates N "R1" words in array B.

Subroutine BDNYQ (OMEGA,DB,PHI,N)

BDNYQ is the QR subroutine that creates Bode and Nyquist on the SC4020. Section 4.7.6 of Volume I provides additional information on the QR frequency response option.

Subroutine BEGIN

BEGIN initializes default values for QR. See sec. 6.3.1 of Volume I for additional details.

Subroutine BIGGER

BIGGER is used to extend dynamic data arrays in QR.

BLOCK DATA

BLOCK DATA initializes all named common blocks in QR.

Subroutine BRITEV

BRITEV sets the SC4020 exposure to heavy. In addition, entry FAINTV is included in this subroutine to set the exposure to light.

Subroutine BUFFIL

BUFFIL takes the current set of SC4020 instructions and packs them into the TAPE99 output buffer OUTI. When OUTI is filled to 408 CDC words, subroutine BUFOUT is called to empty the buffer and reset the buffer count pointer.

Subroutine BUFOUT

BUFOUT performs a BUFFER OUT to transfer labeled common block OUTI, array IOUT, to disk unit 99. Once the BUFFER OUT operation is completed, counters are set to reflect that the disk unit 99 buffer is empty.

BUFOUT is called to empty the buffer and reset the buffer count pointer.

Subroutine CANCEL

CANCEL takes the user designated POLE(S) and ZERO(S) arrays and determines if pole-zero pairs can cancel. The subroutines prints cancelled pole-zero pairs, remaining zeros and polynomial coefficients.

Subroutine CCOMPE (A,NA,CX,CY)

CCOMPE evaluates a polynomial with complex coefficients given in the array A. The complex value returned is CY for the complex point CX.

Subroutine CHESS (F,MS,B)

CHESS forms the upper hessenburg matrix of the input matrix F, which is MS by MS. The hessenburg form is stored over the input matrix F. B is a scratch complex vector.

Subroutine CIRC1V (L,RI,RO,AI,DI,KA,KODE,KDO,AMM)

CIRC1V's purpose is to draw circular grids on the SC4020.

CALL CIRC1V (L,RI,RO,AI,DI,KA,KODE,KDO,AMM)

INPUT:

L	Frame advance control. ≤0 do not advance frame. ≥0 advance the frame.
RI	Value in radial units of the center of the displayed grid (radial offset).
RO	Value in radial units of the outside of the grid.
AI	Angle to be displayed at the 3 o'clock position on the grid, modulo 360 (Angular offset).
D1	In a counterclockwise direction the angle difference used for display of radial lines. D1 = 0 no lines drawn.
KA	The direction of plot angles. = 1 The angles appear in a counter clockwise direction. = 1 The angles appear in a clockwise direction.
KODE	Grid type control. = 1 Circular. = 2 Left semicircle displayed at right hand edge. = 3 Right semicircle displayed at left hand edge. = 4 Top semicircle displayed at bottom edge. = 5 Bottom semicircle displayed at top edge. = -1 Circular with entire grid within margins. Allows truncation of the grid if the margins do not specify a square space.
KDO	Grid Control. = 0 only the outer most circle will be displayed with no labeling. ≠ 0 Full gradation will occur along with full labeling.

AMM Raster increment between minor arcs usually between 8. and 30 (See DXDYV).

OUTPUT: None

STATUS
AFTER CALL:

Scale factors will be set up for a polar mapping onto a cartesian grid. center of the plot of this grid is at point (0,0) and the outside of the circle is at a distance (RO-RI) from the center. In the case of a circular plot, the point (0,0) will be at the center of the space set up by the margins; in the case of a semicircle, it will be at the center of the margin at the open end. In all cases, the angle that will be mapped at the 3 o'clock position is 0^o = (Al- Al).The space is set up so the angles when mapped will be ascending in the clockwise direction for clockwise space; descending in the clockwise direction for a counterclockwise space. All radii drawn will be labeled. To suppress radial lines, set D1 = 0.

Subroutine CLDIV (A,N,B,C,M)

CLDIV divides a complex polynomial by a linear expression (Z/B). A is array of complex coefficients; N is degree of polynomial; B is complex constant in linear factor (Z + B); C is complex array containing quotient and remainder. C(1) contains the remainder and C(2) through C(N + 2) contain the coefficients of the quotient polynomial contained in C(2) through C(M + 2).

Subroutine CLEAR (M,N)

CLEAR zeros N elements in array M. Entry point NEGMAT reverses the sign of N elements in array M and entry BLKFLL fills N elements in array M with blank characters. If N is less than or equal zero, no action is taken.

Subroutine CLIM (U,MATORD,MATDE1,II,JJ,KK,ROW,COL)

CLIM performs gaussian elimination by rows or columns. Entry point GAUSS assumes the pivotal element has been identified.

Subroutine COFSIN (II,ROW)

COFSIN recalulates the leading coefficient of the matrix polynomial based on the number of permutations of rows or columns.

Subroutine COL

COL calculates the branch flag ICOMTY. This branch flag is then used in the QR main program.

Subroutine COLROW (U,MATORD,MATDE1)

COLROW is called when the user has requested a PRINT PARTIAL MATRIX input option (see sec. 6.3.2 of Vol. I).

Subroutine CQREIG (A,N,ROOT,V,U)

CQREIG computes the eigenvalues through use of the QR similarity transformation. A is an N by N matrix; ROOT is the eigenvalue array; and V and U are scratch vector arrays.

Subroutine CRAMER (U,VECTOR,MATORD,MATDE1)

CRAMER substitutes the VECTOR array into a user defined column of the matrix polynomial U. MATORD and MATDE1 are dimension specifications.

Subroutine CREAD (U,MO,MD,V,ME,IROW,ICOL)

CREAD performs the input options of CONTINUOUS MATRIX, DELETE, DELETE AND REDUCE, MATRIX, MATRIX COMPLEX, REPLACE, REPLACE COMPLEX, TRUNCATE, and TRUNCATE and REDUCE. Further information on these program options can be obtained from section 6.3.2 of Volume I.

Subroutine CREV (P,NP,PR)

CREV reverses the order of coefficients of a complex polynomial in an array. P is the array of polynomial coefficients which will be reversed; N is degree of polynomial stored in P; and PR is array of polynomial coefficients stored in reverse order.

Subroutine CSAVE (U)

CSAVE saves the matrix polynomial on disk unit 90 after certain program options have been executed. For example, after data associated with the MATRIX input option has been read by QR, subroutine CSAVE is called. In addition, CSAVE contains the entry point CSAVE2. This section of the subroutine saves the matrix polynomial on disk unit 91. CSAVE2 is called at the beginning of an eigenvalue rooting.

Subroutine DXDYV (1,XL,XR,DX,N,I,NX,DC,IERR) or (2,YB,YT,DY,M,J,NY,DC,IERR)

A call to this subroutine will help provide an optimum calling sequence for GRID1V. Output from DXDYV will be DX,N,I,NX, and IERR or DY,M,J,NY, and IERR. Additional information on XL,XR,YB,YT,DX,CY,N,M,I,J,NX, and NY is located with Subroutine GRID1V. DC limits the density of the grid. The grid lines drawn by GRID1V, using arguments supplied through DXDYV, will be no closer than DC raster positions. DC should never be less than 3.0. DC = 8.0 or 20.0 are recommended values. IERR is an error indicator. If it is non-zero, then subsequent use of GRID1V will probably not produce a satisfactory grid with the given arguments.

Subroutine DRWGRD (IOMGAR,IOMGAL,DBTOP,DBBOT)

DRWGRD creates the grid for bode frequency plots based on abscissa and ordinate minimums and maximums specified by the user.

Subroutine EIGMAT (U, MATORD, MATDE1, ID, ROW, COL, GROW, GCOL, IPIV, ROOT)

EIGMAT sets eigenvalue rooting constants and calls a number of subroutines to place matrix polynomial in normalized form.

Subroutine ELMCHK (U,MATORD,MATDE1,II,JJ,KK,ROW,COL)

ELMCHK scans the matrix for reducible rows or columns. U is the matrix array. MATORD is number of rows in U; MATDE1 is highest power of S plus 1. II, JJ and KK identify the row, column and S + 1 pivotal location for reduction. ROW and COL are logical arrays to keep track of which rows and columns have been scanned. ELMCHK is also entered at COLCHK and POLCHK.

Subroutine EPLUSD (U,MATDURD,MATDE1)

EPLUSD adds two matrices. It is used when calculating unsteady frequency response.

Subroutine EQVCOL (N,IDIM,A,ISTACK,LOWST,LOC,IEQUIV,INIX)

EQVCOL detects and isolates equivalence classed under reachability within a graph, given a connection matrix of the graph.

Subroutine ETIMEC (AK2,VK)

ETIMEC forms a new matrix from a straight-forward multiplication.

Subroutine EXPAND (U,MATORD,F,DF,ISIZE,IORD)

This subroutine expands a matrix polynomial that has second, third, etc. order terms (S^2, S^3, . . .) to a first order system (S^1 terms only).

Subroutine FINDPV (U,MATORD,MATDE1,II,JJ,RDW,COL)

FINDPV finds the complex element along a given row or column that has the largest magnitude.

Subroutine FLUTEI (AK1,AK2,U,MX,NX)

FLUTEI determines the two k-value matrices that must reside in core for the proper interpolation. The matrix polynomial corresponding to the user's input k-value is then formulated, rooted, and printed. In addition, eigenvalues and k-values are placed on disk (TAPE96) for plotting.

Subroutine FORMPM (ROOT,NROOT,FUNCT,RFUNCT,COEF,NPOLY)

FORMPM forms a polynomial array of NROOT eigenvalues, the array ROOT. COEF is the coefficient of the highest power of S.

Subroutine FRAMEV (N)

FRAMEV advances the SC4020 microfilm, places the data, page number and titles on the new grid.

Subroutine FRGRID

FRGRID reads input data associated with the PLOT FREQUENCY RESPONSE and GUST SPECTRUM commands. Section 6.3.6 of Volume I contains information about those data cards. FRGRID also has the entry point gust.

Subroutine FRLINE (X,Y,M,SYMB)

FRLINE is the same as PRLINE except that it is used in the COMPUTE FLUTTER 2 overlay, QRU.7.5.

Subroutine GAINS

GAINS reads input data associated with the GAINS and DC GAIN commands. Section 2.3.3., Vol. I, has additional information on these data cards. GAINS also has the entry point GAINDC.

Subroutine GAUSPM (U,MATORD,MATDE1,MATONE)

GAUSPM determines the highest power of S that is used in a matrix polynomial.

Subroutine GRID1V (L,XL,XR,YB,YT,DX,DY,N,M,I,J,NX,NY)

This routine will produce a grid which has some lines emphasized and some lines labeled, as desired. (Margin spaces will be reserved at the top, left side, and bottom of the grid. These margins will normally be 24 raster counts wide.) The routine sets exposure to heavy. At the completion of the routine, scale factors will have been established and made available, for the conversion of floating point values to raster coordinates, during the operation of subsequent sub-programs in the package.

L = 0 The film will be advanced, ID and corner marks will be printed.

L = 1 The film will be advanced, the date and frame count will be placed in the upper right-hand corner of the frame. Corner marks will be suppressed.

L = 2 The film will not be advanced and the identification information will not be displayed.

L = 3 The film will be advanced but the identification information will be suppressed. Corner marks are printed.

L = 4	The film will be advanced and both the ID and the corner marks will be suppressed.
XL,XR	Floating point values of X for the left-most and right-most limits of the grid.
YB,YT	Floating point values of Y for the bottom and top limits of the grid.
DX,DY	Floating point data increments at which vertical and horizontal grid lines, respectively, will be displayed. If DX = 0, then no vertical grid lines will be shown. To prevent crowding, no more than 120 grid lines in one direction will be drawn.
N,M	Integers that cause every Nth vertical grid line and every Mth horizontal grid line to be retraced for emphasis. If N = 0, then no vertical line is emphasized. If M = 0, then no horizontal line is emphasized. If either N or M is negative, a square grid will be drawn. Note that if N = 0 or M = 0 and a square grid is required, then both N and M should be made negative.
I,J	Integers that cause every Ith vertical grid line and every Jth horizontal grid line to be labeled. If I = 0 then no vertical line is labeled. If J = 0 then no horizontal line is labeled. Note that label margin space is in addition to the margin reserved for titles.
NX,NY	Integers indicating number of characters to be displayed in the labels of vertical and horizontal lines respectively. A positive sign will give a label in decimal format. A decimal point must be counted as one of the characters, but the sign is not counted. The largest number of digits permitted is 6; i.e., NX(or NY) can be + 7 at most. A negative sign will produce a label in scientific notation (e.g., 1.25×10^{2}). The sign, decimal point, and exponent will be displayed in addition to NX (or NY) characters. NX (or NY) must be no less than -6.

Subroutine GMARGN (N)

GMARGN calculates the gain margin of an input frequency response.

Subroutine HELPER (U,MATORD,MATDE1,MATSIZ,ROW,COL,ADDRT,TEMP)

HELPER will compress the input matrix polynomial by: removing rows or columns that are zero except for one element, removing rows or columns with like powers of S, removing rows or columns with a common polynomial factor. Additional information on these reduction techniques is available in reference 2.

Subroutine INLAP (NT)

INLAP calculates the time response for an expression in Laplace transform with multiple poles. Immediate calculation by residue method is performed for single poles, while for the remaining poles, calculation is done by solving a linear system of equations. Further technical information on INLAP is presented in section 4.8.1, Volume I. NT is the number of time history data points.

Subroutine IORESP (U,CVEC,IPR,MATORD,MATDE1,AK2,SO,OMEGA, DB,PHI,VECTOR,CVECT)

IORESP calculates the unsteady frequency response.

Subroutine IRSCAL (A,T,IA,N,M)

gIRSCAL provides scaling for irreducible submatrices.

Function IXV(X)

IXV converts a floating point abscissa value X to a fixed point raster value IX ready to plot. A previous call to XSCALV must have been made before using IX = IXV(X). IX = A*X = B.

Function IYV(Y)

IYV is similar to IXV, setting Y for X, IY for IX, C for A, D for B and YSCALV for XSCALV.

Function JJFIND(IPIV,II,N)

JJFIND finds the row/column corresponding to pivot located in column/row II.

Subroutine KGRID (LADV,XL,YB,XR,YT,LOG)

KGRID sets the raster positions for linear or logarithmic plots, calls subroutines DXDYV and GRID1V based on the input LOG (LOG = -1, time response grid; LOG = 0, root locus grid; LOG = 1, frequency response grid). XL and XR are maximum and minimum abscissa values. YB and YT are maximum and minimum ordinate values.

Subroutine KPLOT (X,Y,IPTS,KHAR,XLF,YBOT,XRT,YTOP)

KPLOT plots the character specified in KHAR for the X-Y pairs (IPTS of them) on the SC4020 with the limits of XLF,XRT,YBOT, and YTOP.

Subroutine LABARC (RI,R,DR,N,NL,KDO)

LABARC labels to the right-hand side of the 3, 6, 9, and 12 o'clock positions. The value is in radial units of specified arcs in the polar grid.
Input:

RI The radial offset between the space and the label.

R The radius in the space of the outermost circle in radial units.

DR The increment value in radial units to be used for the label.

N The index to be used for labeling such that every Nth DR will be labeled as RI+N.DR.

NL The number of characters to be used in the label.

KDO Label Control.

= 0 Do not Label.

≠ 0 Label according to scheme every Nth DR will be labeled as RI + N.DR.

OUTPUT: None

Subroutine LABLV (D,IX,IY,NCHAR,NT,NKMAX)

LABLV converts a floating point quantity into a number in BCD format with the decimal point in the proper position, and it displays the BCD number at the raster coordinates specified.

D The floating point quantity to be printed.

IX,IY The raster coordinates which will position the first character of the label. This may be a leading blank. If the quantity to be displayed is negative, then the minus sign will be displayed one character space to the left of (IX,IY).

NCHAR Number of characters to be displayed including leading blanks and the decimal point, if any. NCHAR must be less than 7 or it can be 7 if one of the characters is the decimal point.

NT The number of times each character is to be displayed. NT = 2 will give a darker label.

NDMAX Maximum number of characters to be displayed to left of the decimal point. Before the BCD label is displayed, it is right adjusted so that there will be NDMAX characters in front of the decimal point. These may now include leading blanks.

NOTE: If NCHAR has a negative sign, then the label will be displayed in scientific notation. In this case, NCHAR must be less than 7 and NDMAX can take any value.

If NCHAR = -1 then the display will be of form
Y.x10 ± YY

If NCHAR = -3 then the display will be of form
Y.YYx10 ± YY

If NCHAR = -6 then the display will be of form
Y.YYYYYx10 ± YY

The space required for scientific notation is greater than for the ordinary rotation.

Subroutine LABLXC (T,NX,IX)

LABLXC labels the abscissa axis of a SC4020 plot.

Subroutine LABLYC (T,NY,IY)

LABLYC labels the ordinate axis of a SC4020 plot.

Subroutine LINEV (IX1,IY1,IX2,IY2)

LINEV draws a line between raster coordinates (IX1,IY1) and (IX2,IY2)

Subroutine MATMUL (A,MA,NA,BM,NA,C)

MATMUL performs the matrix multiplication C = A*B.

Subroutine MATPRT (A,M,N,IDENT)

MATPRT prints a matrix (real or complex) that may have row and column dimensions that are unequal. M is number of rows in A. N is number of columns. IDENT is 10 characters of alphanumeric identification.

Subroutine MNSQVS(N)

MNSQVS calculates PSD response, prints frequency and PSD response and plots PSD response. MNSQVS also has the entry point FPRINT.

Subroutine MOMEGA (ROOT,NROOT)

MOMEGA determines the range of frequencies that will be within the unsteady frequency response.

Subroutine MOVEFR (OMEGA1,DB1,PHI1,OMEGA2,CPS2,FMAG2,DB2,PHI2)

MOVEFR moves the calculated set of unsteady frequency response data points to a higher core address in central memory. This action is performed at the end of the QRU.6.3 overlay just before entry into the QRU.6.6 overlay.

Subroutine MSTORE (A,B,N,M)

MSTORE moves N elements of A into B. Entry point SWAP swaps elements of A and B and entry point ROTSM stores B into A by decrementing.

Subroutine MULLP (C,N,B,BX,ANS,IERR,RES)

MULLP finds all roots or a single root of a polynomial with complex coefficients. The roots are found by Muller's method of parabolic iteration with deflation.

C = an array containing coefficients of the polynomial.
N = degree of polynomial.
ANS = complex root array.

Subroutine MUSER (A,M,N,IMAT)

MUSER is a portion of the user-supplied FORTRAN interface code. Section 6.8 of Volume I provides additional information on its use.

Subroutine NODEC (XLOW,XUP,YLOW,YUP,NX,NY)

NODEC determines the number of decimal places for grid labeling numbers.

Subroutine NORMAL (U,MATORD,ROW,COL,GROW,GCOL,IPIV)

NORMAL normalizes the matrix polynomial using Gaussian elimination. This subroutine has the same effect as multiplying the matrix polynomial by the inverse of the leading matrix.

Function NXV (X)

NXV is a function subprogram that converts a floating point abscissa value X to a fixed point raster value NXV, ready to plot. A previous call to XSCALV must have been made before using IX = NXV(X). If X lies outside the limits, then IX = 0, otherwise, NXV = A*X+B.

FUNCTION NYV (Y)

NYV is similar to NXV except a fixed point raster is calculated for the ordinate direction.

Subroutine ORDER (ROOT,NROOT)

Subroutine ORDER orders NROOT eigenvalues, ROOT array, in descending complex absolute value order. If the ordering occurs during a COMPUTE FLUTTER option, eigenvalues are ordered according to flutter velocity.

Subroutine PEAKS (N)

PEAKS calculates the minimums and maximums of an input frequency response.

Subroutine PGSET (U,MO,MD,PHASE,GAIN)

PGSET calculates a complex value of gain given a gain and phase value. This complex gain value is multiplied times each U matrix element that is designated as a gain location. Section 6.3.3 of Volume I has additional detail on root locus.

Subroutine PLOTTM (TIME,GAIN,N)

This subroutine plots time responses generated by INLAP onto the SC4020. In addition, the plot vectors associated with time response are placed on disk unit 92.

Subroutine PLOTV (IX,IY,NS)

This routine plots at point (IX,IY) the character represented by NS. The following table shows the possible characters that PLOTV can plot.

NS	CHAR	NS	CHAR	NS	CHAR	NS	CHAR
0	0	16	+	32	_	48	
1	1	17	A	33	J	49	/
2	2	18	B	34	K	50	S
3	3	19	C	35	L	51	T
4	4	20	D	36	M	52	U
5	5	21	E	37	N	53	V
6	6	22	F	38	O	54	W
7	7	23	G	39	P	55	X
8	8	24	H	40	Q	56	Y
9	9	25	I	41	R	57	Z
10	0	26	π	42	•	58	°
11	=	27	.	43	$	59	,
12	"	28	)	44	*	60	(
13	'	29	β	45	γ	61	'
14	δ	30	±	46	~	62	Σ
15	α	31	?	47	cl	63	•

Subroutine PLTLI (N,X,Y,JX,JY,IERR)

This routine joins N points (X,Y) by straight lines. If points lie off the grid area, a line will be drawn to the intersection(s) of the grid boundary and the line if this intersection(s) exists.

N = Number of points to be plotted.

X,Y = Arrays containing the X and Y floating points to be plotted.

JX = X skip factor

JY = Y skip factor

IERR = Error location which, at the end of the program, will contain the number of points outside the plotting area.

Note that scale factors must be established before using this program. This can be done, for example, by a previous call to GRID1V.

Subroutine PMRGIN (N)

PMRGIN calculates the phase margin of an input frequency response.

Subroutine POINTD (IX,IY, ± NS)

POINTD plots a symbol, according to the value of NS, at the point specified by (IX,IY).

IX,IY = Raster coordinates at the point to be plotted, fixed point.

±NS = NS is an integer which selects the plotting symbol. If NS is positive an extra plotting dot will also be plotted.

The same table of symbols is used by both POINTV and POINTD.

Note: This routine should be used whenever possible in place of POINTV(IX,IY,NX,ANY).

NS	Symbol	NS	Symbol	NS	Symbol	NS	Symbol	NS	Symbol	NS	Symbol	NS	Symbol
0	•	9	⊡	±18	■	27	S	36	D	45	2	-11	C
1	⊙	10	H	±19	⊘	28	W	37	A	46	3	-12	V
2	X	11	€	±20	⊠	29	Q	38	B	47	9	-30	(
3	⊡	12	V	±21	≁	30	(.	39	E	48	8	-32	)
4	Y	13	∫	±22	θ	31	~	40	F	-1	0	-35	▪
5	◆	14	Z	±23	ϕ	32	.)	41	M	-3	□	-36	D
6	•	±15	●	±24	✱	33	N	42	P	-7	L	-47	7
7	•	±16	■	25	I	34	T	43	R	-8	U		
8	⊔	±17	●	26	G	35	⌶	44	Y	-9	▯		

Subroutine POINTV (X,Y, ± NS) or (IX,IY,INS,ANY)

POINTV plots a symbol, according to the value of NS, at the point specified by (X,Y) or (IX,IY). If floating point arguments are used and the point lies outside the plotting area, no symbol is plotted.

X,Y = Floating point data values of the point to be plotted. Scale factors must previously have been established. This can be done, for example by a previous use of GRID1V.

IX = Raster coordinates of the point to be plotted, fixed point.

IY = Note: IX ≠ 0, IY ≠ 0

±NS = NS is an integer which selects the plotting symbol. If NS is positive, an extra plotting dot will also be plotted.

ANY = This is a dummy argument.

Note that the table included in POINTD shows the symbols obtained by various values of NS

Function POLAR (RI,A1,R1,AA,Y,X)

POLAR returns the value in the cartesian space of the position of a given polar point. POLAR also has the entry point POLER.

INPUT:

RI	=	Radial offset in radial units.
AI	=	Angular offset in degrees.
R1	=	Input radius in radial units.
AA	=	Input angle in degrees.

OUTPUT:

Y	=	Vertical value of the point in the cartesian space.
X	=	Horizontal value of the point in the cartesian space.
Y	=	(R1 - RI) SINE (AA-A1)
X	=	(R1 - RI) COSINE (AA - A1)

Note: Polar does not do any plotting

Subroutine PRECON (A,B,D,IA,IB,IC,ID,NSM,M)

PRECON scales the matrix A for hessenberg transformation and QR rooting.

Subroutine PRINTV (N,BCDTXT,IX,IY) or (N,nH....,IX,IY)

PRINTV prints N characters in typewriter mode starting at point (IX,IY). Writing will take place in horizontal rows, 128 characters in a row and 64 rows to a frame. Care must be taken to ensure that the writing does not run off the frame as it will continue at the top left-hand corner of the same frame.

N	=	Number of characters to print.
BCDTXT	=	An array containing the BCD test to be printed, read in with an A type format.
nH....	=	A Hollerith argument containing the text to be printed.
IX,IY	=	The raster coordinates for the center of the first character to be printed, remembering that each character occupies 8 horizontal raster positions and 16 vertical raster positions.

Subroutine PRLINE (X,Y,M,SYMB)

PRLINE is part of the printer plot package in QR and is called by PRPLOT. The purpose is to scan the X-Y vector and determine if any points would fall onto the current line that is to be printed.

Subroutine PRPLOT (X,Y,M,XP,YP,MP,XZ,YZ,MZ,XL,XR,YT,YB,ITYPE)

PRPLOT is called when a printer plot is to be generated for time response, root locus, or V-g analysis.

Subroutine PTPLOT (X,Y,N)

PTPLOT is called when a time response printer plot is to be generated.

Logical Function QCKROO (GAIN,PHASE)

QCKROO calculates and roots a polynomial formed from the ZEROS, POLES and gain of a transfer function.

Subroutine QQQQ7 (V,MATORD,MATDE1,VN,MATSIZ,IROW)

QQQQ7 reforms the V array into VN based on the number of permutations noted in array IROW. This subroutine is called after a DELETE AND REDUCE or TRUNCATE AND REDUCE input option.

Subroutine QRCOST

QRCOST calculates the CRU cost of the QR execution and writes the collected information onto a permanent file.

Logical Function QRROOT (U)

QRROOT sets up a complete real or complex matrix polynomial eigenvalue rooting. U is the starting location of the matrix for which eigenvalues will be found.

Subroutine RADI1V (A1,D1,R,DR,LRAD)

RADI1V draws the radii lines of the polar grid at specified angles, with the option of labeling. For labeling, RADI1V assumes the space is counterclockwise. It will label as though the space is clockwise or counterclockwise from control information. Set the space clockwise after the call if the desired space is clockwise.

INPUT:

A1 = value of the angle to be displayed at the 3 o'clock position module 360.

D1 = Increment to the next angle to be labeled or drawn in the counterclockwise direction.

R = Outer-most point of the radii drawn. Distance from the center in radial units.

DR = Inner-most position for drawing the radii.

LRAD = Labeling control.
= 0 No labels will be shown.
≠ Labels will appear at the outside of every radii.

Note: If D1 = 0 no radii will be drawn.

Subroutine RDMAT (U,MATORD,A,MAT2)

RDMAT sets up calls to the subroutine READTP

Subroutine RDVEC (V,MATORD)

RDVEC sets up calls to the subroutine READTP to read input vectors.

Subroutine READEM (U,MATORD,MATDE1,VECTOR)

READEM reads and processes information associated with READ MATRIX TAPE, READ VECTOR TAPE, and SPACE TAPE. Entry READFT reads k-value matrices and formulates the matrix polynomials corresponding to each individual k-value. Additional information on the COMPUTE FLUTTER options can be found in section 6.3.7 of Volume I.

Subroutine REGRUP (U,MATORD,MATDE1,UU,MATSIZ,ROW,COL)

REGRUP packs the U matrix into the forward part of UU. The rows and columns corresponding to ROW(I) or COL(J) equal ".TRUE." are eliminated.

Subroutine RESP (OMEGA,FMAG2,DB2,PHI2,CPS2)

RESP calculates the Db and phase angle of a polynomial ratio (ZEROS(s)/POLES(s)) for the input value of omega (radians). If QR is calculating a PSD response, the effects of the gust spectrum are included in the calculations.

Subroutine RHESS (A,IC,IA,N)

RHESS transforms an upper block triangular matrix into hessenburg from using direct reduction with pivoting. The upper portion of A is replaced by the hessenburg form and transformation information replaces the lower portion of A.

Subroutine RLGRID

RLGRID reads data associated with the PLOT ROOT LOCUS data card. Section 6.3.3, Volume I explains additional detail for this data card.

Subroutine ROOTS (U,MATORD,MATDE1,II,JJ,ROOT,NR,RAD)

ROOTS obtains the roots of matrix element II, JJ. The leading coefficient is multiplied by the highest power of S.

Subroutine RPRINT (ROOT,PHA,GAIN,NROOT,NGNCK)

RPRINT prints the eigenvalues residing in the ROOT array.

Subroutine RQREIG (A,ROOT,IC,NSM,MM)

RQREIG computes the eigenvalues of the hessenberg matrix A using the QR algorithm. See reference 3 for additional information.

Subroutine RSTORE (A)

RSTORE reads the matrix file that has been written by subroutine CSAVE. The subroutine is called when the QR program requires that the A matrix be read from disk and placed in core. RSTORE also has the entry point RSTOREZ.

Subroutine RUW (IFOR,IVR,NRWA10)

RUW is used to read all interactive data in the QR program.

Subroutine RWA10 (ICARD,N)

RWA10 reads and writes N alphanumeric CDC words into array ICARD. Entry point RA10 reads N words, entry point WA10 writes N words.

Subroutine RWE12 (A,NCARD,NVARIB,NDIM)

RWE12 reads and writes variables in format 6E12.5.

Subroutine RWI5 (N,I1,I2,I3,I4,I5,I6)

RWI5 reads and writes variables in I5 format.

Subroutine SCALE1 (Y,NP,YPMAX,YPMIN)

SCALE1 finds linear scaling values that will produce SC4020 plots with convenient X or Y axis increments.

Subroutine SCALE2 (Y,NP,YPMAX,YPMIN)

SCALE2 finds linear scaling values that will produce printer plots with convenient X or Y axis increments.

Subroutine SCAMB

SCAMB is part of the interactive error recovery capability of QR. If an error occurs while interactive input data is being decoded from alphanumeric to binary, SCAMB is entered, an error flag is set, and control is returned to the subroutine RUW.

Subroutine SENSOR (U,MATORD,MATDE1,PHI,NX)

SENSOR is entered with the SENSOR and BODY STATIONS control cards. Section 6.3.2 of Volume I has additional information on these program options.

Subroutine SETMIV (MTL,MTR,MTB,MTT)

SETMIV allows a programmer to alter the standard margin assignments used in GRIDIV. The standard values are 24 raster counts at the top, left and bottom of the grid and 0 at the right.

MTL	Width of area for left margin, in raster counts
MTR	Width of area for right margin, in raster counts
MTB	Width of area for bottom margin, in raster counts
MTT	Width of area for top margin, in raster counts

This routine makes it possible to provide margin space for multiple lines of printed titles and headings. GRIDIV guarantees that these margin values, assigned in SETMIV will not be decreased although they may be increased if necessary.

Subroutine SETORD (U,MATORD,MATDE1,IORD,ID,ISIZE)

SETORD determines the apparent state variable order of matrix polynomial U. In addition the columns of U are arranged to form the first order state variable matrix.

Subroutine SORT (X,Y,N)

This routine sorts the vector X into descending order. If two X elements are interchanged, the corresponding Y elements are also interchanged. There are N elements.

Subroutine STUFF (B,C,D,AI,TR,BR,BS,CS,AIS,TRS,AIF,MATORD,MATSIZ, MDIFF,MORIG,ISLIDE,GROW)

STUFF packs the B, C, D, AI, TR, and BR matrices into the front of the U array.

Subroutine SUBJOM (SO,UO,U1,U2,N)

SUBJOM is used in the unsteady frequency response option. The SO (N x N) matrix is formed by substituting a value of frequency, j, into the matrix polynomial $U2*S^2$ + U1*S + UO. S is the complex differential operator.

Subroutine SVFORM (A,MATORD,MATDE1,ID,ROW,COL,GROW,GCOL,IPIV)

SVFORM transforms the A matrix into state variable form.

Subroutine TAPDIM (NARRAY,NSIZE,MATDIM)

TAPDIM is called immediately after a call to subroutine READTP. TAPDIM checks for subscript and array length sizes.

Subroutine TPRINT (N)

TPRINT controls the detail of time response printing. See section 6.3.5 of Volume I for additional detail.

Subroutine TRANS (A,MATORD,MATDE1,VECTOR,LENGTH)

TRANS is used in the unsteady frequency response option. It transposes the A matrix if all of the gains are along one row and builds the vector of knowns for solution in IORESP.

Subroutine UGMARG (OMEGA,GDB,PHI)

UGMARG is the same as GMARGN except that it is used with the unsteady frequency response.

Subroutine UNSQVS (OMEGA,CPS,FMAG,DB,PHI,PSD,PERCNT)

UNSQVS is the same as MNSQVS except that is is used with the unsteady frequency response. UNSQVS also has the entry point UPSP.

Subroutine UPEAKS (OMEGA,G,ANGLE)

UPEAKS is the same as PEAKS except that it is used with the unsteady frequency response.

Subroutine USERSB (A,B,M,N,INSE)

USERSB is a portion of the user-supplied FORTRAN interface code. Section 6.8 of Volume I provides additional information on its use.

Subroutine VECTRC (IX1,±IDX,IY1,±IDY)

VECTRC can be used in place of LINEV when IDX and IDY are less than 64 rasters each. Since it will result in significant savings of CPU time, use of VECTRC is recommended when there are several short lines to draw.

Subroutine VPRNTD (N,BCDTXT,IX,IY) or (N,nH...,IX,IY)

VPRNTD does not use the typewriter mode, but prints N characters starting at point (IX,IY). Writing will take place in vertical columns. The arguments are the same as described for PRINTV.

Subroutine VREAD (VEC,MATORD,MATDE1)

VREAD is called when VECTOR or REPLACE VECTOR data cards are being processed by QR. Section 6.3.2 of Volume I provides additional detail on these data cards.

Subroutine VUSER (B,M,N,IVEC)

VUSER is a portion of the user-supplied FORTRAN interface code. Section 6.8 of Volume I provides additional information on its use.

Subroutine WARN (NUM,ID)

WARN is called when QR has an indication there may be more elements in an array than program defaults allow.

Subroutine WRITER (A,N,K)

WRITER writes elements of the A array row by row. N rows are written. K represents the power of S associated with the A matrix.

Subroutine WRITUM (U,MATORD,MATDE1)

WRITUM sets up calls to the subroutine WRITER. Entry point PNCH punches the U matrix in QR MATRIX input format. Section 6.3.2 of Volume I explains the QR PUNCH format.

XSCALV will compute scale factors for the X coordinate and will save these for use in other subroutines.

XL = Value of X at left-most limit of the scaled plotting area, floating point.

XR = Value of X at right-most limit of the scaled plotting area, floating point.

ML = Margin space, in raster counts, to be saved at left of grid.

MR = Margin space, in raster counts, to be saved at right of grid.

Only linear scaling is available.

Subroutine YSCALV (YB,YT,MB,MT)

YSCALV will compute scale factors for the Y coordinate and will save these for use in other subroutines.

YB = Value of Y at the bottom limit of the scaled plotting area, floating point.

YT = Value of Y at the top limit of the scaled plotting area, floating point.

MB = Margin space, in raster counts, to be saved at bottom of grid.

MT = Margin space, in raster counts, to be saved at top of grid.

Logical Function ZFIND (U,MATORD,MATDE1)

ZFIND modifies elements in the U matrix polynomial array so that the matrix when rooted will represent the open loop zeros. Sec. 4.2 of Volume I explains the automatic formation of poles and zeros.

3.0 EXTENT OF CHECKOUT

The following matrix displays program options used by the QR test cases.

OPTION	CASE 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AIRFORCE COEFFICIENTS	x															
BEGIN	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BODY STATIONS																x
CANCEL				x								x				x
COMPUTE								x			x	x	x			
COMPUTE CONVOLUTE				x												
COMPUTE FLUTTER	x															
COMPUTE FLUTTER AND ROOT LOCUS	x															
COMPUTE FREQUENCY RESPONSE				x										x		x
COMPUTE FLUTTER 2			x													
COMPUTE LOCUS												x				
COMPUTE POLES				x	x									x	x	
COMPUTE PSD				x											x	
COMPUTE ROOT LOCUS				x					x	x	x	x		x		x
COMPUTE ROOT LOCUS WITH PREVIOUS POLES												x				
COMPUTE ROOT LOCUS WITH PREVIOUS ZEROS												x				
COMPUTE ROOT LOCUS WITH PREVIOUS POLES AND ZEROS												x				
COMPUTE TIME RESPONSE				x											x	
COMPUTE TIME RESPONSE WITH TAU				x												
COMPUTE UNSTEADY FREQUENCY RESPONSE		x														
COMPUTE ZEROS				x										x	x	
CONTINUOUS													x			
CRAMER REPLACE				x				x						x	x	
DC GAIN				x												
DELETE					x											
DELETE AND REDUCE					x											
DIFFERENTIATE				x												
EXECUTE PLOTS												x				
EXECUTE PLOTS WITH POLES												x				
EXECUTE PLOTS WITH ZEROS												x				

OPTION	CASE 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EXECUTE PLOTS WITH POLES AND ZEROS												x				
FORCING FUNCTION				x												
FORM POLYNOMIAL				x								x				
FORM RATION OF POLYNOMIALS				x										x	x	
FREQUENCY RESPONSE		x		x							x			x	x	x
GAINS		x		x					x	x	x	x		x		x
GUST SPECTRUM													x		x	
INTEGRATE				x											x	
MATRIX				x		x	x	x	x	x		x		x	x	
MATRIX COMPLEX																
NO FORM POLYNOMIAL																
PLOT FLUTTER	x	x														
PLOT FREQUENCY RESPONSE		x									x	x		x	x	x
PLOT PSD				x											x	
PLOT ROOT LOCUS	x			x							x	x		x		x
PLOT TIME RESPONSE				x											x	x
PRINT					x	x	x					x	x			
PRINT PARTIAL MATRIX				x												
PTITLE				x								x				x
PUNCH																
READ MATRIX TAPE	x															x
READ VECTOR TAPE																
REPLACE	x	x	x				x					x			x	x
REPLACE COMPLEX							x						x			
REPLACE VECTOR													x			
SENSOR																x
SPACE TAPE																x
STATE				x												
STOP																x
TRUNCATE AND REDUCE						x										
TIME INTERVALS				x											x	x
TIME RESPONSE				x											x	
TRUNCATE						x										
TITLE				x										x		x
UNSTEADY AIRFORCE COEFFICIENTS		x	x													
VECTOR		x	x						x					x	x	
CALL MUSER											x					
CALL USERSB																
CALL VUSER																

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124
May 1977

REFERENCES

1. Miller, R. D.; Kroll, R. I.; and Clemmons, R. E.: Dynamic Loads Analysis System (DYLOFLEX) Summary. Volume I: Engineering Formulation. NASA CR-2846-1, 1979.

2. Richardson, T. M.: A Method for Transforming Differential Equations to State Variable Form. Seattle University Paper, May 1972.

3. Computer Journal. Volume II, number 1, May 1968, pp. 112-114.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-2862		
4. Title and Subtitle: Linear Systems Analysis Program, L224(QR). Volume II: Supplemental System Design and Maintenance Document		5. Report Date: November 1979
		6. Performing Organization Code
7. Author(s): K. W. Heidergott		8. Performing Organization Report No.: D6-44471
9. Performing Organization Name and Address: Boeing Commercial Airplane Company, P.O. Box 3707, Seattle, Washington		10. Work Unit No.
		11. Contract or Grant No.: NAS1-13918
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, D.C. 20546		13. Type of Report and Period Covered: Contractor report May 1975-May 1977
		14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Boyd Perry III

Final Report

16. Abstract

This document describes the computer program known as QR. Classical control systems analysis and synthesis (root locus, time response, and frequency response) can be performed using this program. Volume I discusses program usage and includes a limited theoretical discussion. Volume II describes programming details of the QR program.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
QR; Root Locus; Phase Locus; Frequency Response; Time Response; PSD (Power Spectral Density)	Unclassified - unlimited Subject Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	141	$7.25

*For sale by the National Technical Information Service, Springfield, Virginia 22151

END

April 20, 1981